Business Data Mining (IDS 572)

(source: r-bloggers.com)

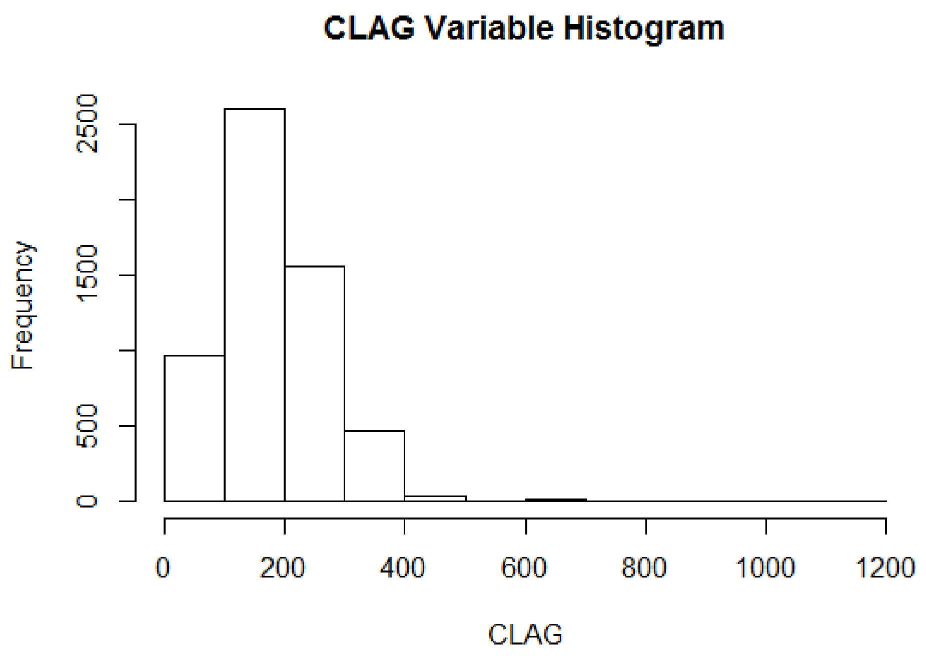
Detecting and removing outliers

In statistics, a outlier is defined as a observation which stands far away from the most of other observations. Often a outlier is present due to the measurements error. Therefore, one of the most important task in data analysis is to identify and (if is necessary) to remove the outliers.

There are different methods to detect the outliers. Below we provide a few of these methods. To illustrate these methods we use the "hmeq" data set which can be found on blackboard.

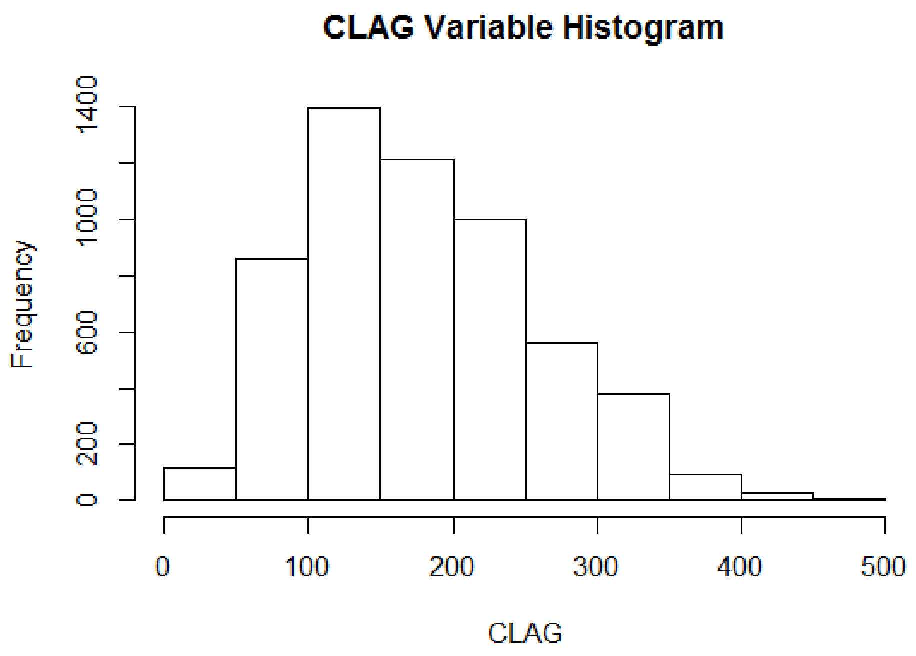
Method 1: To detect the outliers, you can first draw the histogram to determine the range of outliers.

```
> hist(hmeq$CLAG, main = "CLAG Variable Histogram", xlab = "CLAG")
```



As you can see there are outliers for the CLAG variable. To remove the outliers we can use the "subset(DataSet_name, Variable_name < Bound)" function similar to the following code:

```
> DataNew = subset(hmeq, CLAG < 500)
> hist(DataNew$CLAG, main = "CLAD Variable Histogram", xlab = "CLAG")
```

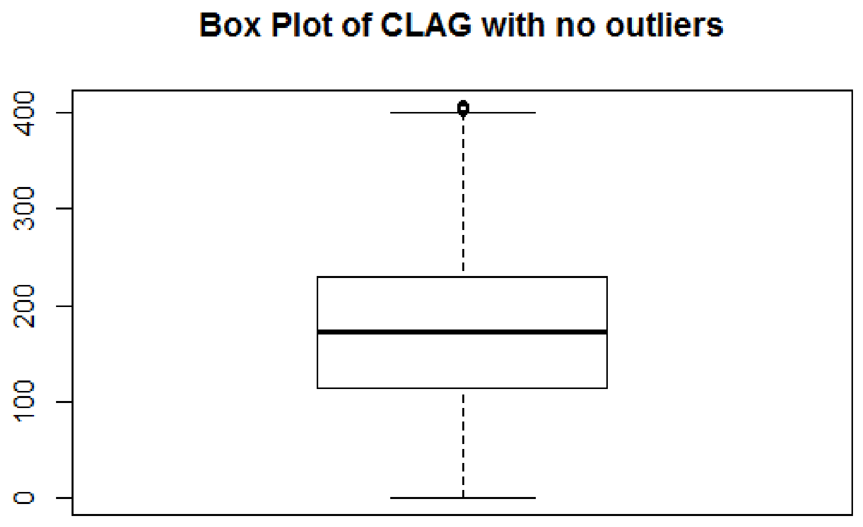


If you have more than one variable with outliers you can use the following formula:

```
> NewData = subset(Data_name, Var1_name < Bound1 & Var2_name < Bound2 & ···)
```

Method 2: To detect the outliers, the command "boxplot.stats()$out" can be used which uses the Tukey's method to identify the outliers ranged above and below the $1.5 \times IQR$.

```
> OutLiers = boxplot.stats(hmeq$CLAG)$out   # We first save all the outliers in the vector OutLiers
> CLAGnoOut = ifelse(hmeq$CLAG % in % OutLiers, NA, hmeq$CLAG)   # if the value of CLAG is in OutLiers then we replace it by NA (or any other value)
> boxplot(CLAGnoOut)
```



Handling missing values

In R, missing values are represented by the symbol NA (not available) . Impossible values (e.g., dividing by zero) are represented by the symbol NaN (not a number).
To test if there is any missing values in data, we can use the function "is.na()" which returns TRUE for each missing value.

```
> sum(is.na(hmeq$NINQ))  # This give you the number of missing values in the variable NINQ
[1] 510
```

To count the number of rows where one or more columns contain NA (incomplete cases), we can use "sum(!complete.cases())".

```
> sum(complete.cases(hmeq$NINQ))  # Count of complete cases in the variable NINQ
[1] 5450
> sum(!complete.cases(hmeq$NINQ))  # Count of complete cases in the variable NINQ
[1] 510
> which(!complete.cases(hmeq$NINQ))  # Which cases (row numbers) are incomplete
```

The summary function of a data frame also counts the occurrence of NA in each column.

Replacing NA values

The function “na.omit()” returns the object with listwise deletion of missing values.

```
> NINQ_Imputed = na.omit(hmeq$NINQ)  # Create new variable without missing values
> sum(is.na(NINQ_Imputed))
[1] 0
```

Replacing missing values by a particular value

To replace missing values by a particular value like mean we can use the following code:

```
> hmeq$NINQ[is.na(hmeq$NINQ)] = mean(hmeq$NINQ, na.rm=TRUE)  # Recode all NA
in NINQ as the average value
> sum(is.na(hmeq$NINQ))
[1] 0
```

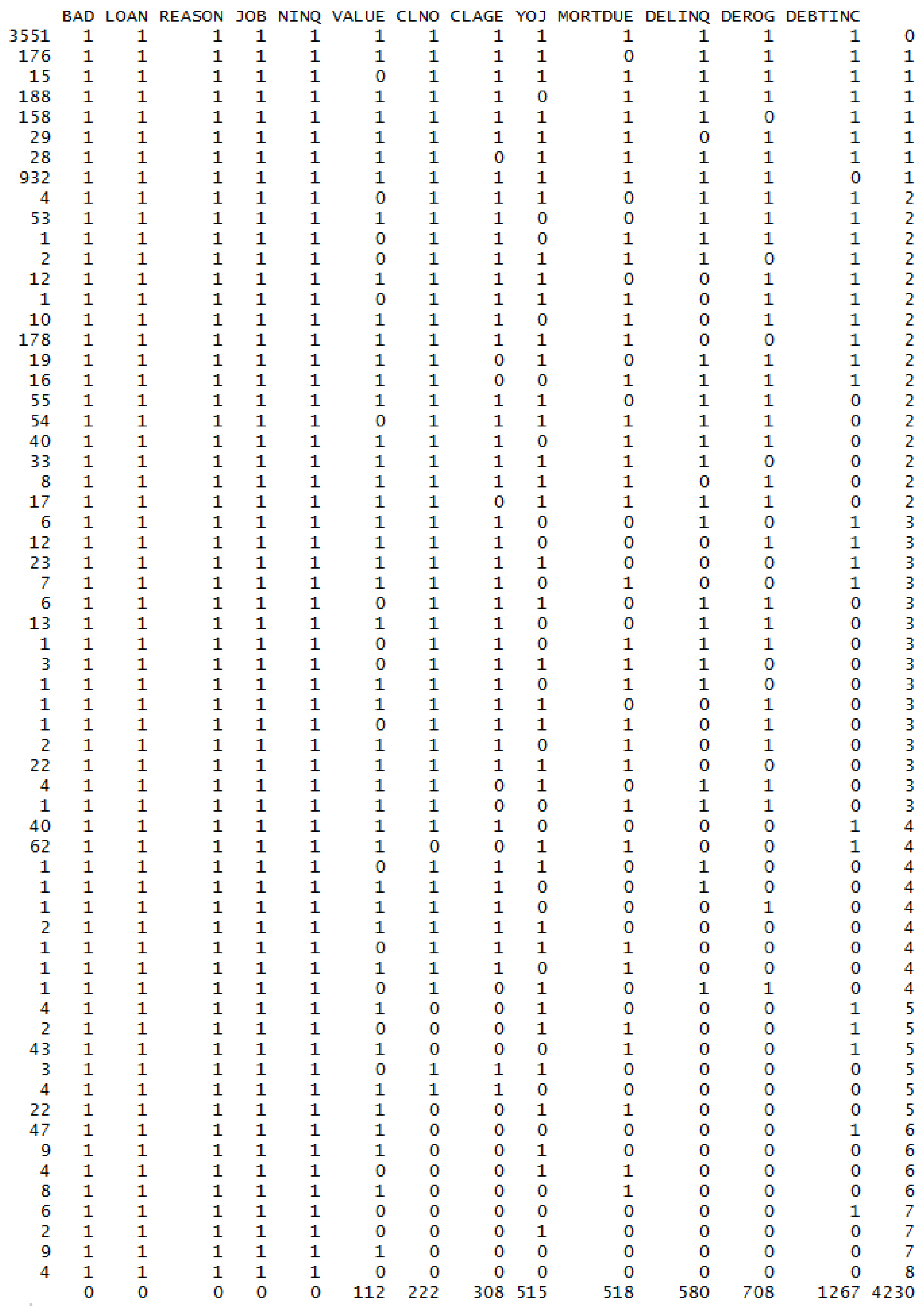
Notice that arithmetic functions on missing values yield missing values. So mean(hmeq$NINQ) returns NA. To remove the missing values in the computation of mean, we should use “na.rm = TRUE”.

While some quick fixes such as mean-substitution may be fine in some cases, such simple approaches usually introduce bias into the data, for instance, applying mean substitution leaves the mean unchanged (which is desirable) but decreases variance, which may be undesirable. The “mice” package helps you imputing missing values with plausible data values. These plausible values are drawn from a distribution specifically designed for each missing datapoint.

Using mice for looking at missing data pattern

The mice package provides a function “md.pattern()” to get better understanding of the pattern of missing data.

```
> library(mice)
> md.pattern(hmeq)
```



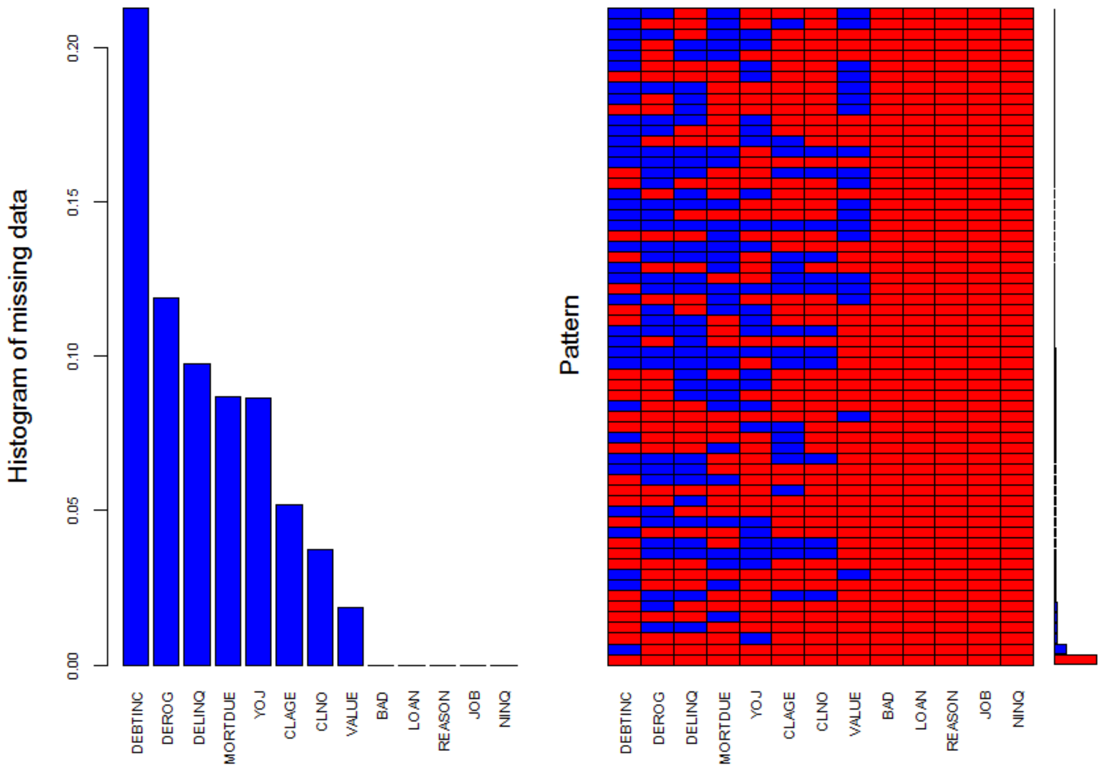
```
     BAD LOAN REASON JOB NINQ VALUE CLNO CLAGE YOJ MORTDUE DELINQ DEROG DEBTINC
3551   1    1      1   1    1     1    1     1   1       1      1     1       1    0
 176   1    1      1   1    1     1    1     1   1       0      1     1       1    1
  15   1    1      1   1    1     0    1     1   1       1      1     1       1    1
 188   1    1      1   1    1     1    1     1   0       1      1     1       1    1
 158   1    1      1   1    1     1    1     1   1       1      1     0       1    1
  29   1    1      1   1    1     1    1     1   1       1      0     1       1    1
  28   1    1      1   1    1     1    1     0   1       1      1     1       1    1
 932   1    1      1   1    1     1    1     1   1       1      1     1       0    1
   4   1    1      1   1    1     0    1     1   1       0      1     1       1    2
  53   1    1      1   1    1     1    1     1   0       0      1     1       1    2
   1   1    1      1   1    1     0    1     1   0       1      1     1       1    2
   2   1    1      1   1    1     0    1     1   1       1      1     0       1    2
  12   1    1      1   1    1     1    1     1   1       0      0     1       1    2
   1   1    1      1   1    1     0    1     1   1       1      0     1       1    2
  10   1    1      1   1    1     1    1     1   0       1      0     1       1    2
 178   1    1      1   1    1     1    1     1   1       1      0     0       1    2
  19   1    1      1   1    1     1    1     0   1       0      1     1       1    2
  16   1    1      1   1    1     1    1     0   0       1      1     1       1    2
  55   1    1      1   1    1     1    1     1   1       0      1     1       0    2
  54   1    1      1   1    1     0    1     1   1       1      1     1       0    2
  40   1    1      1   1    1     1    1     1   0       1      1     1       0    2
  33   1    1      1   1    1     1    1     1   1       1      1     0       0    2
   8   1    1      1   1    1     1    1     1   1       1      0     1       0    2
  17   1    1      1   1    1     1    1     0   1       1      1     1       0    2
   6   1    1      1   1    1     1    1     1   0       0      1     0       1    3
  12   1    1      1   1    1     1    1     1   0       0      0     1       1    3
  23   1    1      1   1    1     1    1     1   1       0      0     0       1    3
   7   1    1      1   1    1     1    1     1   0       1      0     0       1    3
   6   1    1      1   1    1     0    1     1   1       0      1     1       0    3
  13   1    1      1   1    1     1    1     1   0       0      1     1       0    3
   1   1    1      1   1    1     0    1     1   0       1      1     1       0    3
   3   1    1      1   1    1     0    1     1   1       1      1     0       0    3
   1   1    1      1   1    1     1    1     1   0       1      1     0       0    3
   1   1    1      1   1    1     1    1     1   1       0      0     1       0    3
   1   1    1      1   1    1     0    1     1   1       1      0     1       0    3
   2   1    1      1   1    1     1    1     1   0       1      0     1       0    3
  22   1    1      1   1    1     1    1     1   1       1      0     0       0    3
   4   1    1      1   1    1     1    1     0   1       0      1     1       0    3
   1   1    1      1   1    1     1    1     0   0       1      1     1       0    3
  40   1    1      1   1    1     1    1     1   0       0      0     0       1    4
  62   1    1      1   1    1     1    0     0   1       1      0     0       1    4
   1   1    1      1   1    1     0    1     1   1       0      1     0       0    4
   1   1    1      1   1    1     1    1     1   0       0      1     0       0    4
   1   1    1      1   1    1     1    1     1   0       0      0     1       0    4
   2   1    1      1   1    1     1    1     1   1       0      0     0       0    4
   1   1    1      1   1    1     0    1     1   1       1      0     0       0    4
   1   1    1      1   1    1     1    1     1   0       1      0     0       0    4
   1   1    1      1   1    1     0    1     0   1       0      1     1       0    4
   4   1    1      1   1    1     1    0     0   1       0      0     0       1    5
   2   1    1      1   1    1     0    0     0   1       1      0     0       1    5
  43   1    1      1   1    1     1    0     0   0       1      0     0       1    5
   3   1    1      1   1    1     0    1     1   1       0      0     0       0    5
   4   1    1      1   1    1     1    1     1   0       0      0     0       0    5
  22   1    1      1   1    1     1    0     0   1       1      0     0       0    5
  47   1    1      1   1    1     1    0     0   0       0      0     0       1    6
   9   1    1      1   1    1     1    0     0   1       0      0     0       0    6
   4   1    1      1   1    1     0    0     0   1       1      0     0       0    6
   8   1    1      1   1    1     1    0     0   0       1      0     0       0    6
   6   1    1      1   1    1     0    0     0   0       0      0     0       1    7
   2   1    1      1   1    1     0    0     0   1       0      0     0       0    7
   9   1    1      1   1    1     1    0     0   0       0      0     0       0    7
   4   1    1      1   1    1     0    0     0   0       0      0     0       0    8
       0    0      0   0    0   112  222   308 515     518    580   708    1267 4230
```

The output tells us that 3551 samples are complete, 176 samples miss only MORTDUE, 15 samples miss only the VALUE and so on.

To visualize these information, the package VIM can be used.
]

> library(VIM)
> aggr_plot = aggr(hmeq, col = c('red', 'blue'), numbers = TRUE, prop = TRUE, sortVars = TRUE, labels = names(hmeq), cex.axis = 1, gap = 0, ylab = c("Histogram of missing data", "Pattern"))
The color red indicates observed values and the color blue indicates missing values.

```
Variables sorted by number of missings:
Variable      Count
 DEBTINC 0.21258389
   DEROG 0.11879195
  DELINQ 0.09731544
 MORTDUE 0.08691275
     YOJ 0.08640940
   CLAGE 0.05167785
    CLNO 0.03724832
   VALUE 0.01879195
     BAD 0.00000000
    LOAN 0.00000000
  REASON 0.00000000
     JOB 0.00000000
    NINQ 0.00000000
```



The plot helps us to understand which variables has the largest number of missing values. In addition, this plot shows that almost 21% of data are missing the DEBTINC value, 11% are missing DEROG value and so on.

To know more about the details of arguments in aggr() function you can use the help option in R.

Imputing the missing values using mice

The "mice()" function takes care of imputing process.

```
> NewData = mice(hmeq, m=5, maxit=50, meth='pmm', seed=500)
> summary(NewData)
```

A couple of notes on the parameters:

- m=5 refers to the number of imputed datasets. Five is the default value.
- meth='pmm' refers to the imputation method. In this case we are using predictive mean matching as imputation method. Other imputation methods can be used, type methods(mice) for a list of the available imputation methods.

To check the imputed data, for example for CLAGE, we can use the following code"

```
> NewData$imp$CLAGE
```

```
            1          2          3          4          5
4  276.841935  308.75918 114.461955 105.530376 101.82614
11 186.633333  146.93333 217.786841  84.837250 125.76667
18 100.616421   62.05011 134.000074 122.766667  89.43649
22 147.100000  115.60000 177.566667  62.900000 288.16667
52 138.164705  230.06728 297.001608 102.500000 196.55422
64 321.633333  115.90000  55.358803 297.482505 199.73871
74 289.545673  135.57298 310.366757 246.754666  55.55805
93  17.200000   85.25084  17.460750  91.715293 228.03600
```

The output shows the imputed data for each observation (first column left) within each imputed dataset (first row at the top).

To check the imputation method used for each variable, we can use "NewData$meth".

```
> NewData$meth
```

```
    BAD    LOAN MORTDUE   VALUE  REASON     JOB     YOJ   DEROG  DELINQ   CLAGE    NINQ
  "pmm"   "pmm"   "pmm"   "pmm"   "pmm"   "pmm"   "pmm"   "pmm"   "pmm"   "pmm"   "pmm"
   CLNO DEBTINC
  "pmm"   "pmm"
```

We can get back the completed dataset using the complete() function.

```
> complete(NewData, 1)
```

As far as categorical variables are concerned, replacing categorical variables is usually not advisable. Some common practice include replacing missing categorical variables with the mode of the observed ones, however, it is questionable whether it is a good choice. Even though in this case no data points are missing from the categorical variables, we remove them from our dataset (we can add them back later if needed) and take a look at the data using summary().