A04 - Suthi de Silva - CSC 285 - 11th Jan 2024

Scenario

You are working on a research project where you are examining test scores from around the country. Different teachers have sent in their data and they are now located in Percentages.csv (/srv/R/CSC_May21/Data). The data comes from a test that is worth 120 points. The teachers were instructed to send in the test scores as percentages. As the team data scientists, you and your partner have the job of cleaning up the data before it is used in the study. Prepare a document to share with your team at the next meeting, describing what you did to clean the data. Make sure you include reproducible code, because you might need to make changes later!

Loading data

```
percentages <- read_csv("/srv/R/CSC285_public/Suthi /Percentages.csv")
```

Here are the first 20 rows of uncleaned data for your reference.

```
head(percentages, n = 20)
```

```
## # A tibble: 20 x 1
##    Marks
##    <chr>
##  1 41
##  2 14
##  3 93
##  4 -14
##  5 55
##  6 97
##  7 119
##  8 105
##  9 3
## 10 96
## 11 -7
## 12 -4
## 13 32
## 14 112
## 15 99
## 16 71
## 17 -14
## 18 75
## 19 84
## 20 77
```

Cleaning up data

1. Removing '.' and 'NA' from Marks column

```
percentages[150 : 160,]
```

```
## # A tibble: 11 x 1
##    Marks
##    <chr>
##  1 109
##  2 76
##  3 38
##  4 0
##  5 36
##  6 -6
##  7 39
##  8 <NA>
##  9 82
## 10 91
## 11 -4
```

```
percentages[80 : 90,]
```

```
## # A tibble: 11 x 1
##    Marks
##    <chr>
##  1 89
##  2 71
##  3 24
##  4 40
##  5 102
##  6 105
##  7 .
##  8 -19
##  9 84
## 10 112
## 11 105
```

As we could notice that '.' and 'NA' would affect negatively further calculation that would be doing with these data, we would have no option but to remove them.

```
percentages <- percentages[-c(which(percentages$Marks == '.' )), ]
percentages <- percentages[complete.cases(percentages), ]
```

So it would look like below.

```
percentages[150 : 160,]
```

```
## # A tibble: 11 x 1
##    Marks
##    <chr>
##  1 76
##  2 38
##  3 0
##  4 36
##  5 -6
##  6 39
```

```
##  7 82
##  8 91
##  9 -4
## 10 117
## 11 65
```

```
percentages[80 : 90,]
```

```
## # A tibble: 11 x 1
##    Marks
##    <chr>
##  1 89
##  2 71
##  3 24
##  4 40
##  5 102
##  6 105
##  7 -19
##  8 84
##  9 112
## 10 105
## 11 32
```

2. Converting 'char' type data to 'numeric'

Here we could notice that data types of 'Marks' column is 'char', as we had '.' and 'NA' in the data set originally. We should change it to 'numeric', to perform mathematical calculations later.

```
percentages$Marks <- as.numeric(percentages$Marks)
```

Now it should have changed to 'numeric'.

```
str(percentages)
```

```
## tibble [383 x 1] (S3: tbl_df/tbl/data.frame)
##  $ Marks: num [1:383] 41 14 93 -14 55 97 119 105 3 96 ...
```

3. Removing outliers

First let's check whether we have any outliers.

```
percentages[c(which(percentages$Marks < 0 | percentages$Marks > 120 )), ]
```

```
## # A tibble: 54 x 1
##    Marks
##    <dbl>
##  1   -14
##  2    -7
##  3    -4
##  4   -14
##  5   -12
##  6   -17
##  7    -1
##  8   -15
##  9   -18
## 10   -17
## # ... with 44 more rows
```

It appears that there are some outliers, so we would remove them like this.

```
percentages <- percentages[-c(which(percentages$Marks < 0 | percentages$Marks > 120 )), ]
```

Now let's double check the existence of outliers

```
percentages[c(which(percentages$Marks < 0 | percentages$Marks > 120 )), ]
```

```
## # A tibble: 0 x 1
## # ... with 1 variable: Marks <dbl>
```

4. Adding a percentage column

Now let's calculate and add a percentage column next to marks, for each mark student scored.

```
percentages$Percentage <- c(((percentages$Marks)/120)*100)
head(percentages, n = 10)
```

```
## # A tibble: 10 x 2
##    Marks Percentage
##    <dbl>      <dbl>
##  1    41       34.2
##  2    14       11.7
##  3    93       77.5
##  4    55       45.8
##  5    97       80.8
##  6   119       99.2
##  7   105       87.5
##  8     3        2.5
##  9    96       80
## 10    32       26.7
```

5. Adding a unique Score ID for each score

```
percentages$ScoreID <- 1:nrow(percentages)
percentages %>% relocate(ScoreID, .before= Marks)
```

```
## # A tibble: 329 x 3
##    ScoreID Marks Percentage
##      <int> <dbl>      <dbl>
##  1       1    41       34.2
##  2       2    14       11.7
##  3       3    93       77.5
##  4       4    55       45.8
##  5       5    97       80.8
##  6       6   119       99.2
##  7       7   105       87.5
##  8       8     3        2.5
##  9       9    96       80
## 10      10    32       26.7
## # ... with 319 more rows
```

Summary

Here is a summary of the data set we just cleaned up.

```
summary(percentages)
```

```
##      Marks          Percentage        ScoreID
##  Min.   :  0.00   Min.   :  0.00   Min.   :  1
##  1st Qu.: 31.00   1st Qu.: 25.83   1st Qu.: 83
##  Median : 62.00   Median : 51.67   Median :165
##  Mean   : 61.28   Mean   : 51.07   Mean   :165
##  3rd Qu.: 90.00   3rd Qu.: 75.00   3rd Qu.:247
##  Max.   :120.00   Max.   :100.00   Max.   :329
```

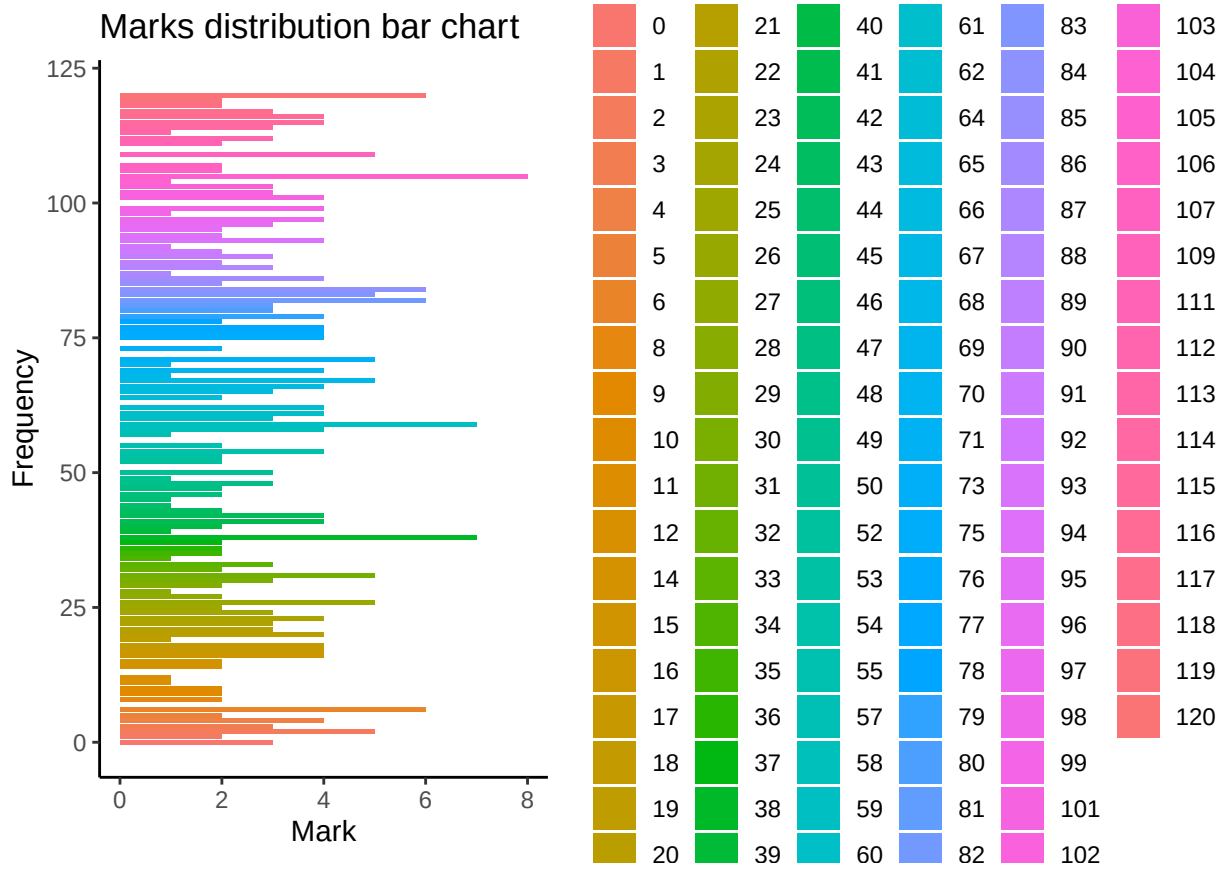
Analysis, and Plots.

We would be only using "Marks" to visualize data as the graphs made from "Percentage" created almost identical looking graphs

Bar plot

We would be making a ar chart to show the frequency or how many times each score has been repeated in the exam.

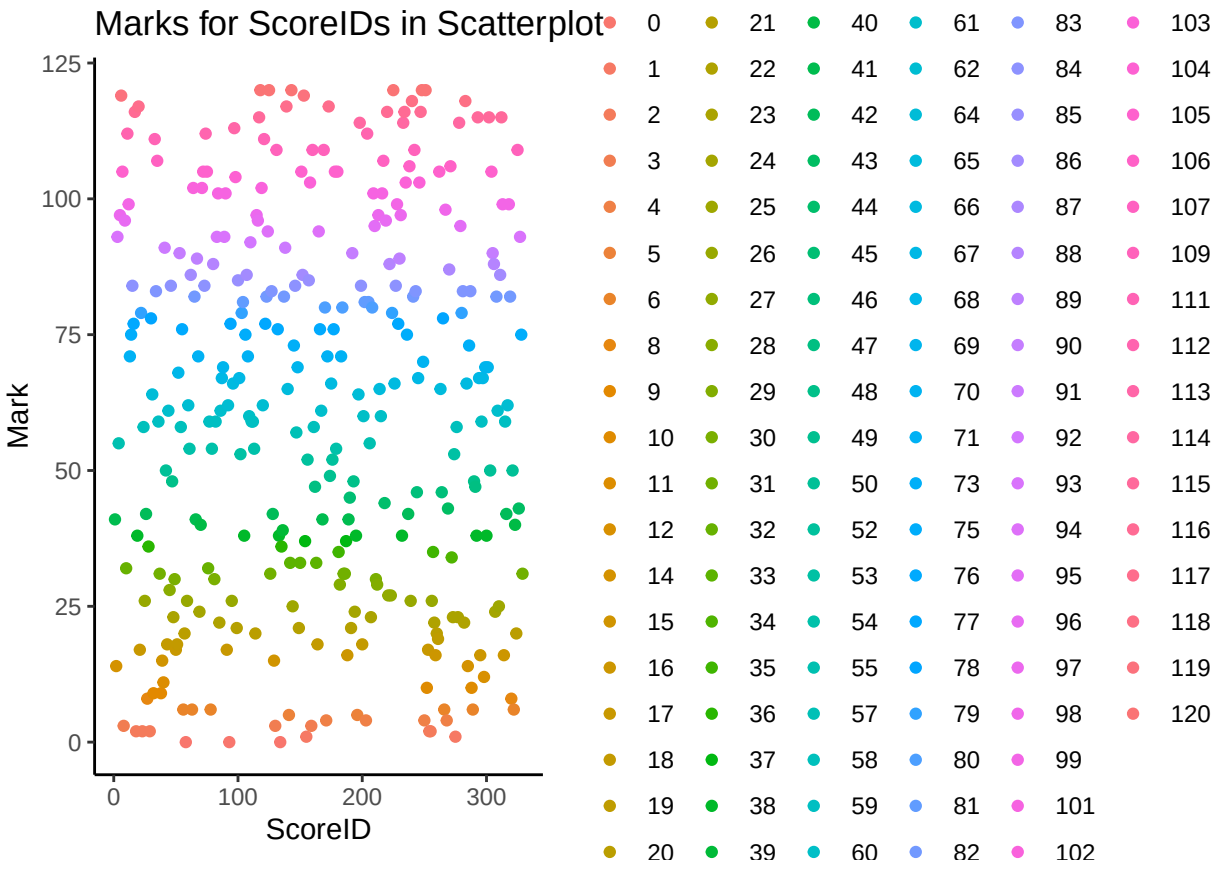
```
ggplot(data = percentages ) +
  geom_bar(mapping = aes( y = Marks, fill = as.factor(Marks)))  +
  labs(title="Marks distribution bar chart",
       x="Mark", y = "Frequency") + theme_classic()
```



Scatter plot

We would be making a scatter plot to show the distribution of scores among the score IDs in the exam. As it would give us a visual explanation how far spread the scores are and find if there is a visible correlation or not. **There is no particular correlation as all the data are spreading all over the graph.

```
ggplot(percentages, aes(x = ScoreID , y = Marks)) +
    geom_point(aes(color = factor(Marks)))  +
  labs(title="Marks for ScoreIDs in Scatterplot",
       x="ScoreID", y = "Mark") + theme_classic()
```



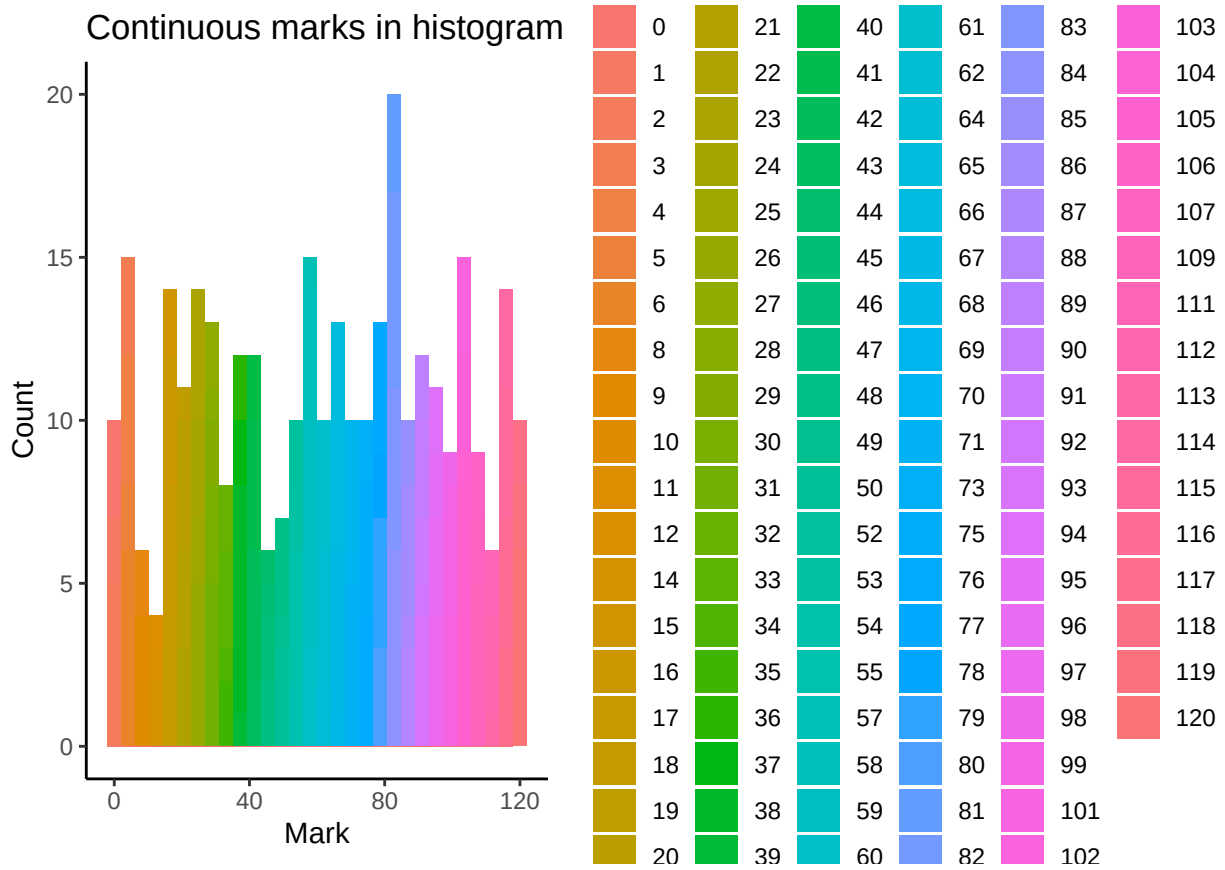
In a sense it feels like ScoreIDs could be the students who gets a certain score.

Histogram

A histogram is used to show the distribution and to summarize discrete or continuous data of marks that are measured on an interval scale.

```
ggplot(data = percentages) +
  geom_histogram(mapping = aes(x = Marks , fill = as.factor(Marks)))  +
  labs(title="Continuous marks in histogram",
       x="Mark", y = "Count") + theme_classic()
```

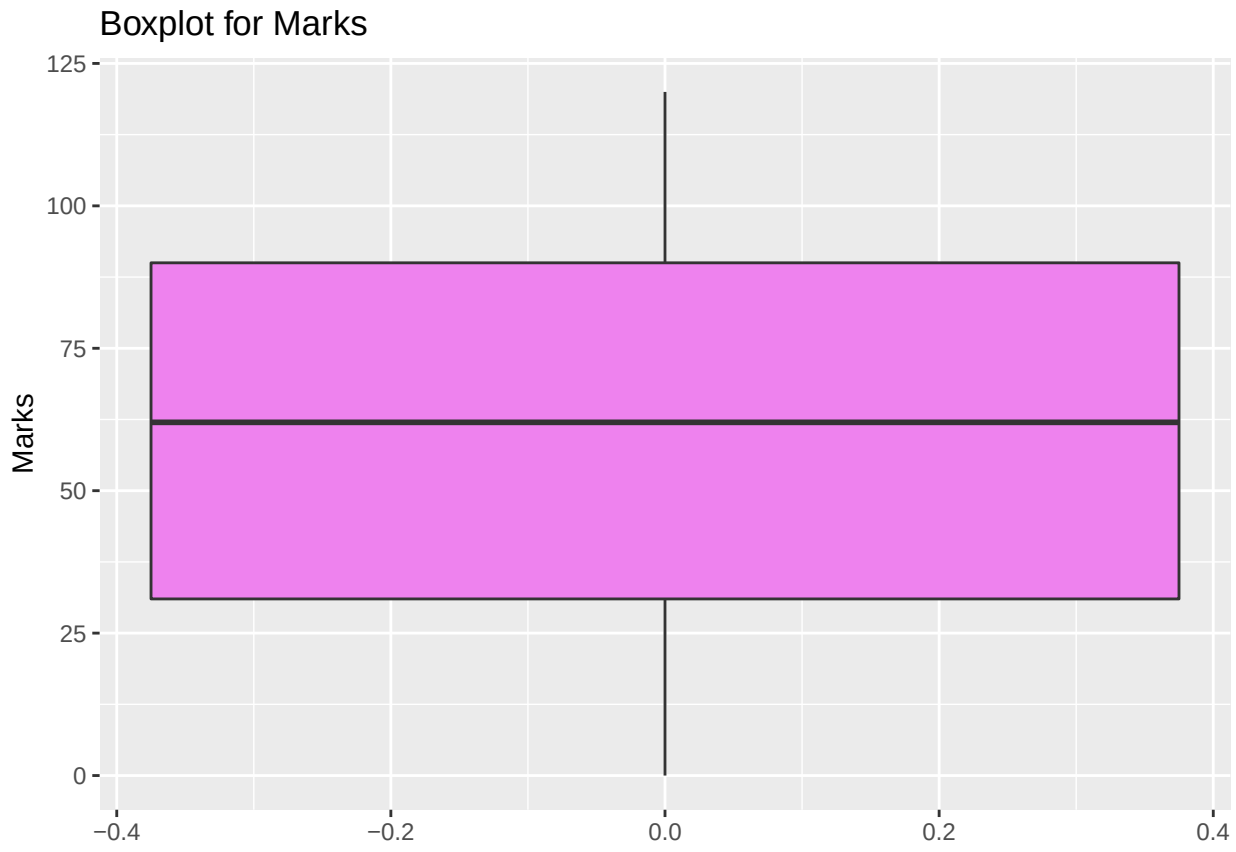
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



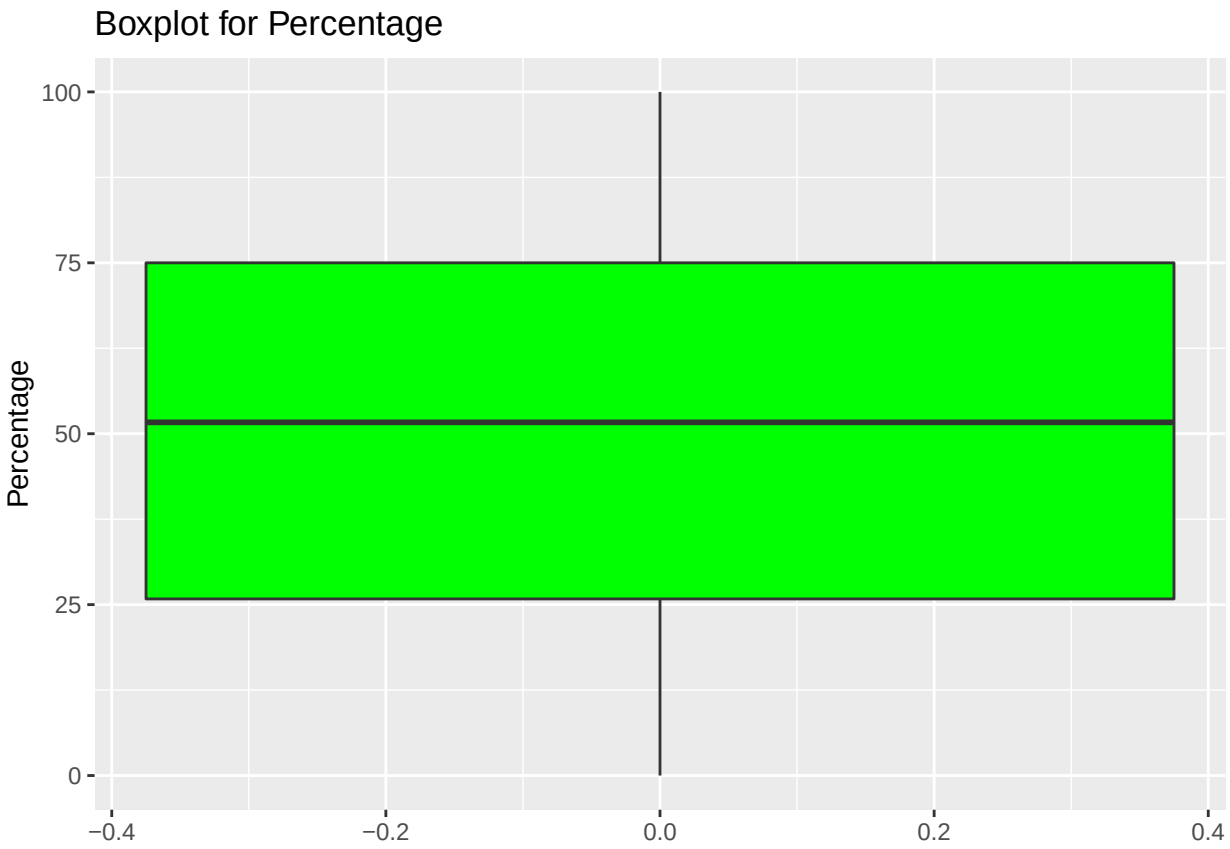
Boxplot

A box plot represents the mean, upper and lower quartiles with interquartile range, min and max of marks and percentages.

```
ggplot(percentages, aes(y = Marks)) +
  geom_boxplot(fill="violet") +
  ggtitle('Boxplot for Marks')
```




```
ggplot(percentages, aes(y = Percentage)) +
  geom_boxplot(fill="green") +
  ggtitle('Boxplot for Percentage')
```


Boxplot for Percentage
Percentage
100
75
50
25
0
−0.4
−0.2
0.0
0.2
0.4