eda-1

October 1, 2023

[1]:
```
import csv
from pandas import read_csv
from pandas.plotting import scatter_matrix
import numpy as np
import matplotlib as plt
```

[2]:
```
file_path = "dataset/glass/glass.data"
names = ["Id", "RI", "Na", "Mg", "Al", "Si", "K", "Ca", "Ba", "Fe", "class"]
dataset = read_csv(file_path, names=names)
```

[3]:
```
print(dataset.shape)
```

```
(214, 11)
```

[4]:
```
print(dataset.head(20))
```

```
    Id       RI     Na    Mg    Al     Si     K    Ca   Ba    Fe  class
0    1  1.52101  13.64  4.49  1.10  71.78  0.06  8.75  0.0  0.00      1
1    2  1.51761  13.89  3.60  1.36  72.73  0.48  7.83  0.0  0.00      1
2    3  1.51618  13.53  3.55  1.54  72.99  0.39  7.78  0.0  0.00      1
3    4  1.51766  13.21  3.69  1.29  72.61  0.57  8.22  0.0  0.00      1
4    5  1.51742  13.27  3.62  1.24  73.08  0.55  8.07  0.0  0.00      1
5    6  1.51596  12.79  3.61  1.62  72.97  0.64  8.07  0.0  0.26      1
6    7  1.51743  13.30  3.60  1.14  73.09  0.58  8.17  0.0  0.00      1
7    8  1.51756  13.15  3.61  1.05  73.24  0.57  8.24  0.0  0.00      1
8    9  1.51918  14.04  3.58  1.37  72.08  0.56  8.30  0.0  0.00      1
9   10  1.51755  13.00  3.60  1.36  72.99  0.57  8.40  0.0  0.11      1
10  11  1.51571  12.72  3.46  1.56  73.20  0.67  8.09  0.0  0.24      1
11  12  1.51763  12.80  3.66  1.27  73.01  0.60  8.56  0.0  0.00      1
12  13  1.51589  12.88  3.43  1.40  73.28  0.69  8.05  0.0  0.24      1
13  14  1.51748  12.86  3.56  1.27  73.21  0.54  8.38  0.0  0.17      1
14  15  1.51763  12.61  3.59  1.31  73.29  0.58  8.50  0.0  0.00      1
15  16  1.51761  12.81  3.54  1.23  73.24  0.58  8.39  0.0  0.00      1
16  17  1.51784  12.68  3.67  1.16  73.11  0.61  8.70  0.0  0.00      1
17  18  1.52196  14.36  3.85  0.89  71.36  0.15  9.15  0.0  0.00      1
18  19  1.51911  13.90  3.73  1.18  72.12  0.06  8.89  0.0  0.00      1
19  20  1.51735  13.02  3.54  1.69  72.73  0.54  8.44  0.0  0.07      1
```

[5]:
```
print(dataset.describe())
```

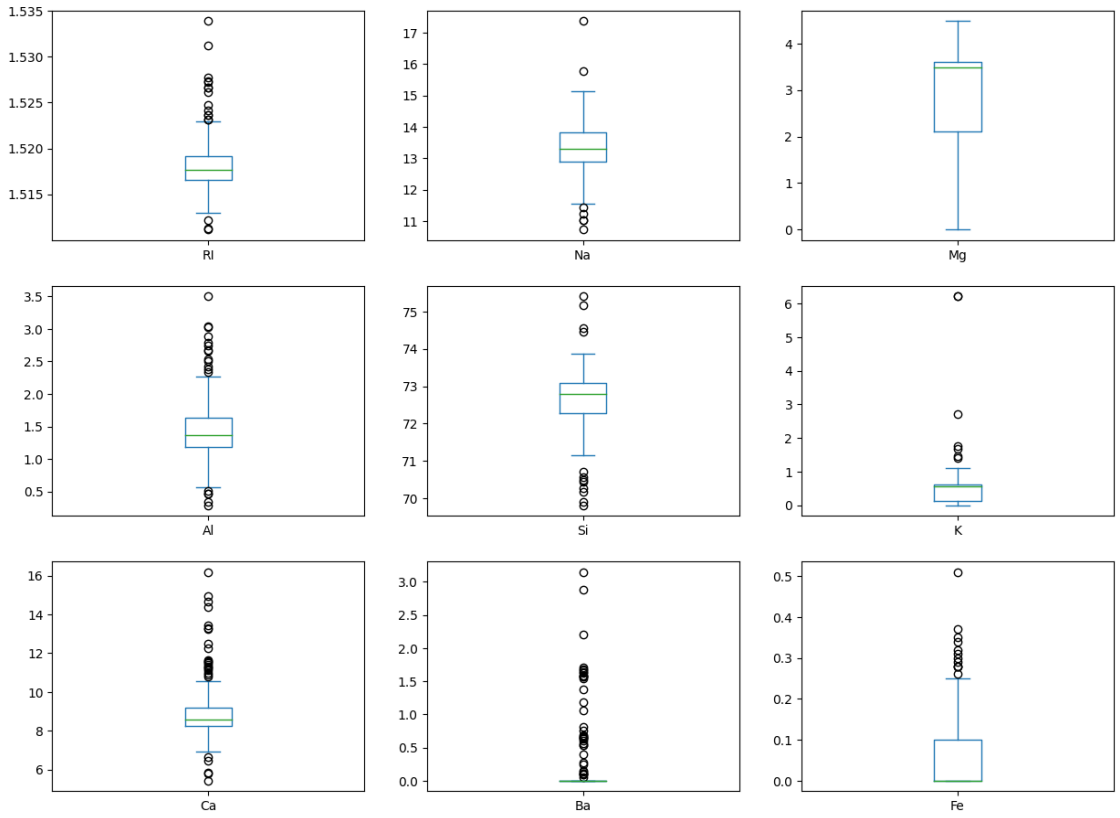
```
               Id          RI          Na          Mg          Al          Si  \
count  214.000000  214.000000  214.000000  214.000000  214.000000  214.000000
mean   107.500000    1.518365   13.407850    2.684533    1.444907   72.650935
std     61.920648    0.003037    0.816604    1.442408    0.499270    0.774546
min      1.000000    1.511150   10.730000    0.000000    0.290000   69.810000
25%     54.250000    1.516522   12.907500    2.115000    1.190000   72.280000
50%    107.500000    1.517680   13.300000    3.480000    1.360000   72.790000
75%    160.750000    1.519157   13.825000    3.600000    1.630000   73.087500
max    214.000000    1.533930   17.380000    4.490000    3.500000   75.410000

                K          Ca          Ba          Fe       class
count  214.000000  214.000000  214.000000  214.000000  214.000000
mean     0.497056    8.956963    0.175047    0.057009    2.780374
std      0.652192    1.423153    0.497219    0.097439    2.103739
min      0.000000    5.430000    0.000000    0.000000    1.000000
25%      0.122500    8.240000    0.000000    0.000000    1.000000
50%      0.555000    8.600000    0.000000    0.000000    2.000000
75%      0.610000    9.172500    0.000000    0.100000    3.000000
max      6.210000   16.190000    3.150000    0.510000    7.000000
```

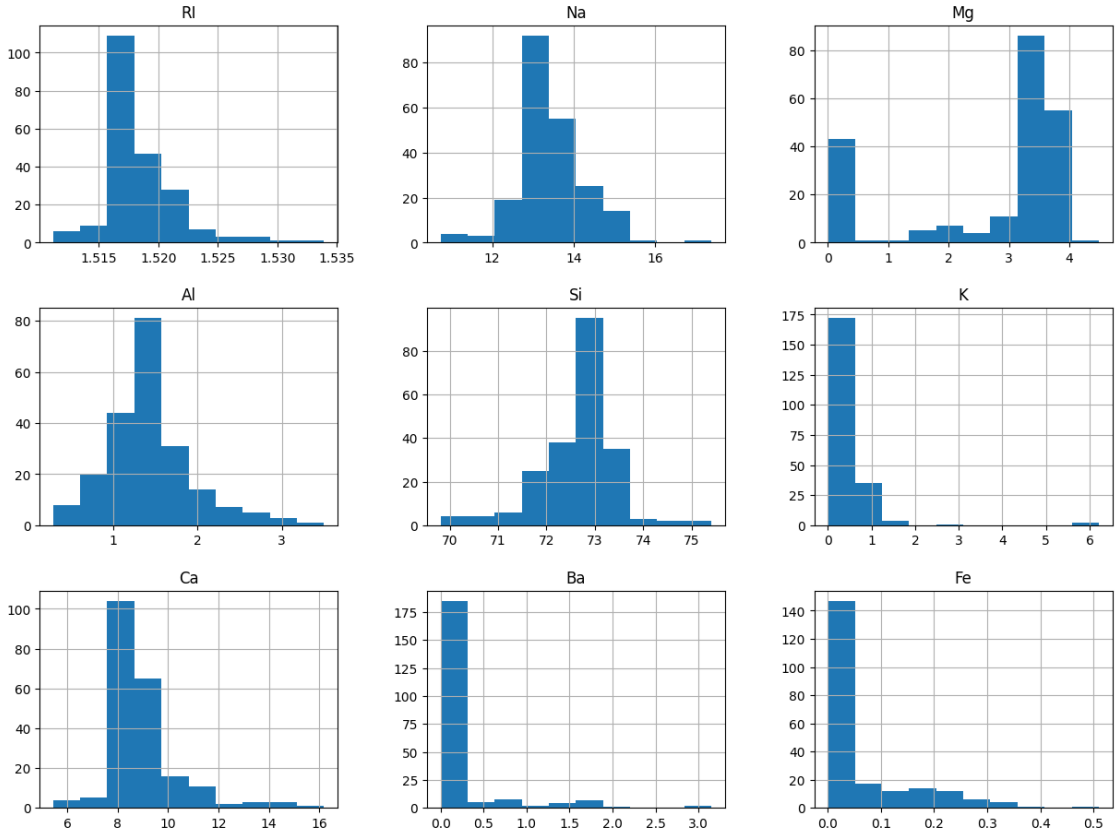
[6]:
```
print(dataset.groupby("class").size())
```

```
class
1    70
2    76
3    17
5    13
6     9
7    29
dtype: int64
```

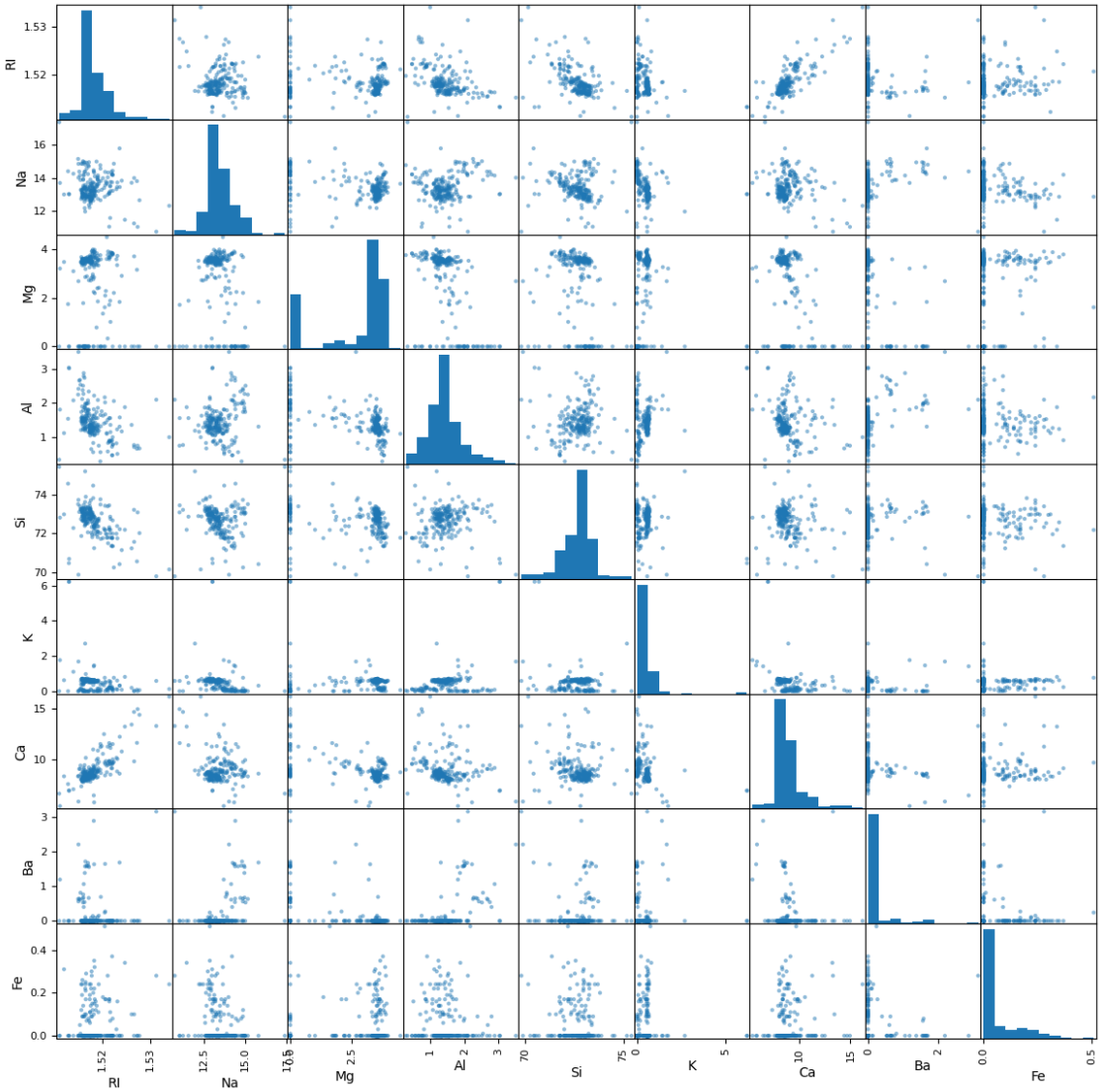
[7]:
```
dataset.drop(columns=['Id', 'class']).plot(kind="box", subplots=True,␣
↪layout=(4,3), sharex=False, sharey=False, figsize=(15,15))
plt.pyplot.show()
```



[8]:
```
dataset.drop(columns=['Id', 'class']).hist(layout=(4,3), figsize=(15,15))
plt.pyplot.show()
```



[9]:
```
scatter_matrix(dataset.drop(columns=['Id', 'class']),figsize=(14,14))
plt.pyplot.show()
```



0.1 Overview

The dataset contains information about different glass types, characterized by various features (such as RI, Na, Mg, Al, Si, K, Ca, Ba, and Fe). The total dataset contains several rows and columns, each row representing a glass sample and each column representing a particular feature or the class of the glass.

0.2 Initial Inspection

Looking at the first 20 rows of the dataset provides a quick overview of the types of data and the range of values present in each feature, helping us identify the nature of information contained in the dataset, and the kind of preprocessing that might be necessary.

0.3 Statistical Summary

A summary statistics analysis reveals insights into the distribution, central tendency, and spread of each feature. It is evident that some features might have different scales, possibly indicating the need for feature scaling for certain machine learning algorithms.

0.4 Class Distribution

Grouping the dataset by class and checking the size of each group allows for understanding the distribution of different types of glass in the dataset. This is crucial in identifying any imbalance in the dataset, which can affect the performance of classification models.

0.5 Boxplots

Boxplots for each feature (excluding 'Id' and 'class') provide visual insights into the distribution and spread of each feature, making it easier to spot outliers or understand the variability of each feature. From the boxplots, it is clear that different features have different distributions and ranges, again signaling the possible need for scaling.

0.6 Histograms

Histograms give a clear picture of the distribution of each feature. The shape and spread of the histograms can give insights into the nature of each feature, showing whether they follow a normal distribution, are skewed, or have a particular pattern.

0.7 Scatter Matrix

Finally, a scatter matrix of the features is generated to visualize the pairwise relationships between the features.

0.8 Conclusion

In conclusion, the exploratory data analysis performed provides significant insights into the Glass Identification Dataset, including the distribution and relationships between different features. These insights are valuable for further data preprocessing steps and in choosing suitable machine learning algorithms for classification tasks. The visualizations, including boxplots, histograms, and the scatter matrix, offer visual cues for understanding the dataset's characteristics and guiding further analysis and model building.