Assignment 2: Data exploration & preparation

31250 Introduction to Data Analytics

Alexander Justin Dealson | 24650196

1A. Initial Data Exploration

1.Attribute Types

Attribute Name: age

Attribute Type: Ratio

Justification: The attribute type of age is ‘Ratio’ because it has a true zero (a new-born) and it can also be said that someone who is 20 years old is twice as old as someone who is 10 years old. In addition, age is a quantitative data type.

Attribute Name: job

Attribute Type: Nominal

Justification: The attribute type of job is ‘Nominal’ because it is a categorical value. The values cannot be compared to one another, and is used to label variables, which has no numeric or value.

Attribute Name: marital

Attribute Type: Nominal

Justification: The attribute type of marital is ‘Nominal’ because it is a categorical value. The values cannot be compared to one another, and is used to label variables, which has no numeric value.

Attribute Name: education

Attribute Type: Ordinal

Justification: The attribute type of education is ‘Ordinal’ because the values can be compared to one another. It represents discrete and the values can be compared to one another.

Attribute Name: default

Attribute Type: Nominal

Justification: The attribute type of default is ‘Nominal’ because it is a categorical value. The values cannot be compared to one another, and is used to label variables, which has no numeric or value.

Attribute Name: housing

Attribute Type: Nominal

Justification: The attribute type of housing is ‘Nominal’ because it is a categorical value. The values cannot be compared to one another, and is used to label variables, which has no numeric value.

Attribute Name: loan

Attribute Type: Nominal

Justification: The attribute type of loan is ‘Nominal’ because it is a categorical value. The values cannot be compared to one another, and is used to label variables, which has no numeric value.

Attribute Name: contact

Attribute Type: Nominal

Justification: The attribute type of attribute contact is ‘Nominal’ because it is a categorical value. The values cannot be compared to one another, and is used to label variables, which has no numeric value.

Attribute Name: month

Attribute Type: Nominal

Justification: The attribute type of attribute month is ‘Nominal’ because it is a categorical value. The values cannot be compared to one another because we cannot deduce that a particular month is earlier in a year than other month if there is no year specified together with the month. In addition, it is used to label variables, which has no numeric or quantitative value.

Attribute Name: day_of_week

Attribute Type: Nominal

Justification: The attribute type of day_of_week is ‘Nominal’ because it is a categorical value. The values cannot be compared to one another, and is used to label variables, which has no numeric value.

Attribute Name: duration

Attribute Type: Ratio

Justification: The attribute type of duration is ‘Ratio’ because a zero can be defined and not arbitrary. Furthermore, it can be said that a duration of 20 seconds is twice as long as 10 seconds. In addition, duration is a quantitative data type.

Attribute Name: campaign

Attribute Type: Ratio

Justification: The attribute type of campaign is ‘Ratio’ because a zero can be defined and not arbitrary. In this attribute, it has true zero which mean there’s a total absence of the variable of interest.

Attribute Name: pdays

Attribute Type: Ratio

Justification: The attribute type of pdays is ‘Ratio’ because a zero can be defined and not arbitrary. In this attribute, it has true zero which mean there’s a total absence of the variable of interest. In addition, pdays is a quantitative data type.

Attribute Name: previous

Attribute Type: Ratio

Justification: The attribute type of previous is ‘Ratio’ because a zero can be defined and not arbitrary. In this attribute, it has true zero which mean zero means none. Furthermore, in this attribute, it is talking about the number of contacts performed before this campaign. By the statement itself, it can be deduced that this attribute is numeric and ratio type. In addition, previous is a quantitative data type.

Attribute Name: poutcome

Attribute Type: Nominal

Justification: The attribute type of attribute poutcome is ‘Nominal’ because it is a categorical value. The values cannot be compared to one another, and is used to label variables, which has no numeric value.

Attribute Name: emp.var.rate

Attribute Type: Interval

Justification: The attribute type of attribute emp.var.rate is ‘Interval’ because it is a numeric data. The values of emp.var.rate can go beyond zero which mean the zero in here does not mean none and it is arbitrary. Moreover, the value can be ordered as the exact differences between the values can be calculated. In addition, if we were to assume this as ratio, it would not make a sense because it cannot be said that a variation of

Attribute Name: cons.price.idx

Attribute Type: Interval

Justification: The cons.price.idx is a measure of inflation, hence it has an interval attribute type due to its nature of having an artificial zero. The CPI can have values of zero or negative which displays the state of the economy.

Attribute Name: cons.conf.idx

Attribute Type: Interval

Justification: The attribute type of attribute cons.conf.idx is 'Interval' because it is a numeric data. The values of cons.conf.idx can go beyond zero which means the zero in here does not mean none and it is arbitrary. Moreover, the value can be ordered as the exact differences between the values can be calculated.

Attribute Name: euribor3m

Attribute Type: Interval

Justification: The euribor3m is the interest rate which is used by European banks for lending purposes. Hence, it may have an artificial zero, meaning that to have zero rate does not mean absence, but is a matter of convenience.

Attribute Name: nr.employed

Attribute Type: Ratio

Justification: The nr.employed obtains ratio attribute type as it shows the number of employees in a bank. In this case, this is a continuous variable that has a natural zero (absence) because the zero here is not another level of scale.

Attribute Name: subscribed

Attribute Type: Nominal

Justification: The attribute type of subscribed is 'Nominal' because it is a categorical value. The values cannot be compared to one another, and is used to label variables, which has no numeric value.

2. Summarising properties for the attributes

Attribute Name: age

Statistics	Value
Mean	40.087
Q1	32
Q2 (Median)	38
Q3	47
IQR	15
Maximum Value	88
Minimum Value	19
Standard Deviation	10.4

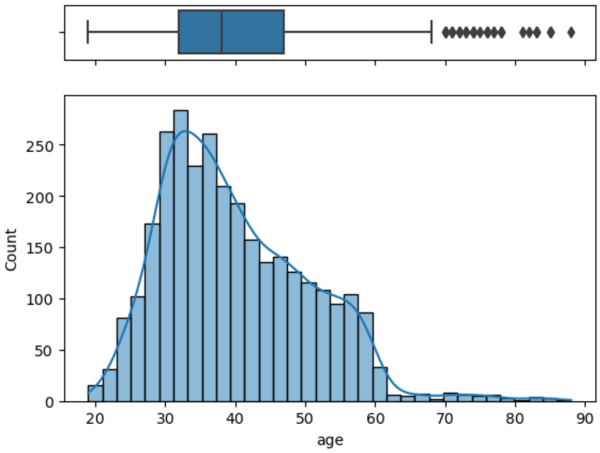
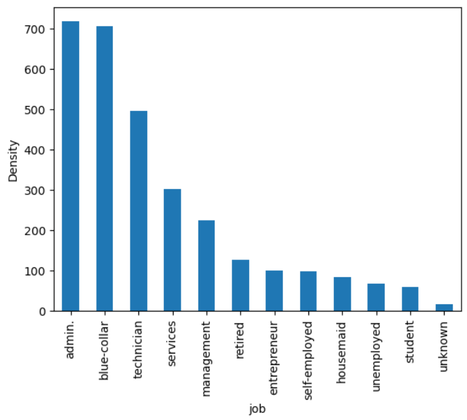


Figure 1. Age Box Plot and Histogram

In Figure 1, the graphs indicate the distribution of age values in datasets. In the given information it was outlined that age refers to the age of the people in this dataset. Utilizing that logic in this data, the dataset has a range from 19 years old (Minimum value) to 88 years old (Maximum value). Through the visualization of box plot, age attribute value is mainly concentrated around 32 to 47 years old. This can also be seen in the histogram and the distribution lines which illustrate a positive skewness for age attributes which mean age values majorly lie in the left side of the histogram. Based on histogram, it can be inferred that the most prominent age group in the sample are people in their 30s whereas people who are 60 years old and older are either outliers or there is only a small number of them. In the Histogram, the curve of the distribution line is more clustered around the mean which implies this dataset has low variations and data are less spread out.

Attribute Name: job

Value	Frequency
admin	718
blue-collar	706
technician	497
service	303
Management	224
retired	126
entrepreneur	101
self-employed	98
housemaid	85
unemployed	67
student	59
unknown (Missing values)	16



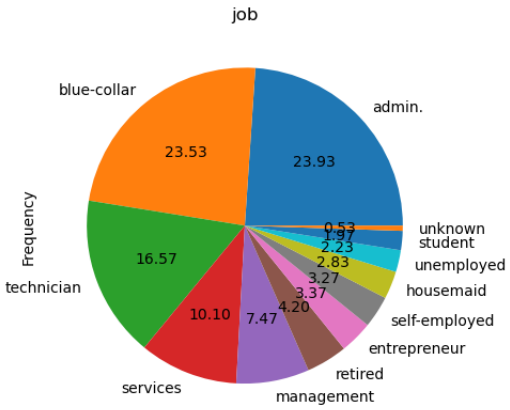


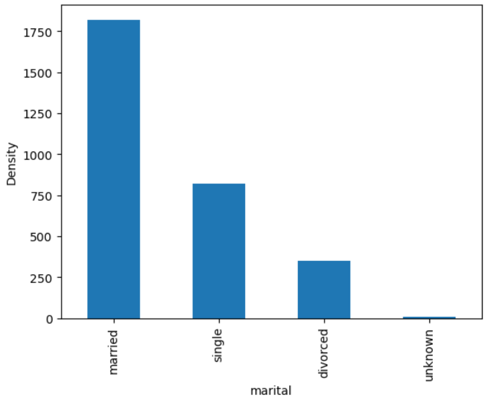
Figure 2. Job Histogram

Figure 3. Job Pie chart

The Histogram above (Figure 2) indicates the several different kinds of jobs and the number of people who work in each kind of job. In the given information it was outlined that job refers to the type of job of the people in this dataset. Utilizing that logic in this data, the mode or most of the people in this dataset work as an ‘admin’ whereas the job which has the smallest number of people is ‘student’. This information can also be seen in the pie chart (Figure 3) where 23.93% of the whole population is ‘admin’ while only 1.97% is a ‘student’. The pie chart consists mostly of ‘admin’ and ‘blue-collar’ which made up to 50% of the pie chart. In addition, we can also see that there’s a small number of missing values about 0.53% identified as ‘unknown’ in this attribute.

Attribute Name: marital

Value	Frequency
married	1828
single	821
divorced	351
unknown	8



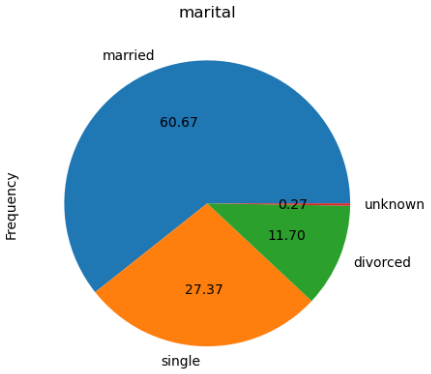


Figure 4. Marital Histogram

Figure 5. Marital Pie chart

The Histogram above (Figure 4) indicates the several different kinds of marital status with their frequency. In the given information it was outlined that marital status refers to the marital status of the people in this dataset. Utilizing that logic in this data, the mode or most of the people in this dataset obtain marital status of ‘married’ and followed by marital status of ‘single’ as the second highest. In contrast, the marital status of ‘divorced’ has the smallest number of people. This information can also be seen in the pie chart (Figure 5) where 60.67% of the whole population is ‘married’ while only 11.70% is ‘divorced’. The pie chart is mostly composed of ‘married’ people. In addition, we can also see that there’s a small number of missing values about 0.27% identified as ‘unknown’ in this attribute.

Attribute Name: education

Value	Frequency
university.degree	964
high.school	665
basic.9y	472
professional.course	400
basic.4y	322
Basic.6y	176
illiterate	1
unknown	119

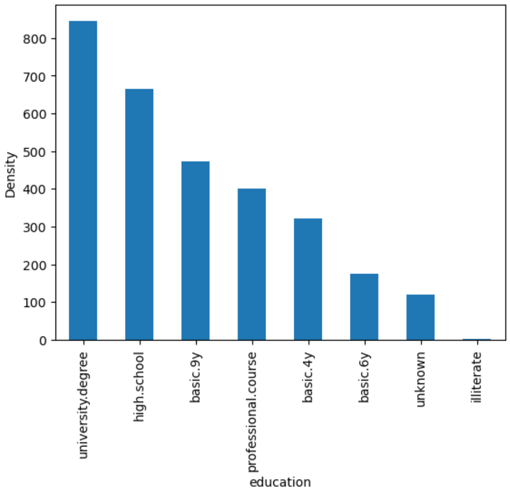


Figure 6. Education Histogram

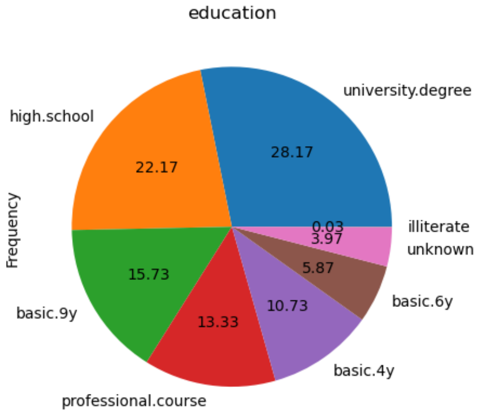


Figure 7. Education Pie chart

The Histogram above (Figure 6) indicates several different levels of education with their frequency. Most of the people in this dataset have a ‘university degree’ and followed by ‘high school’ as the second highest. In contrast, ‘illiterate’ has the smallest number of frequencies of only 1 person. This means it can be a noise or a unique finding. This information can also be seen in the pie chart (Figure 7) where 28.17% of the whole population has a ‘university degree’ while only 0.03% is ‘illiterate’. The pie chart consists mostly of ‘university degree’ and ‘high school’ which made up to 50% of the pie chart. In addition, we can also see that there's a small number of missing values about 3.97% identified as 'unknown' in this attribute.

Attribute Name: default

Value	Frequency
no	2370
unknown	630

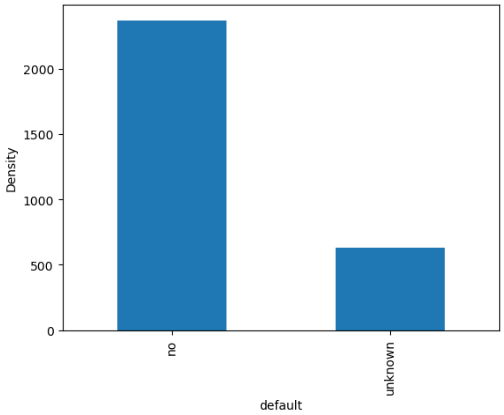


Figure 8. Default Histogram

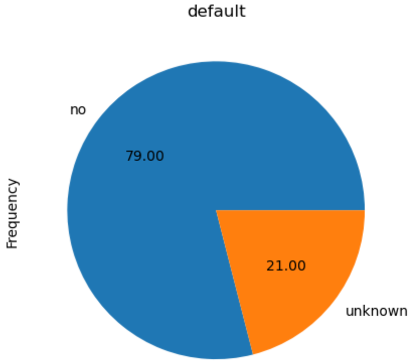
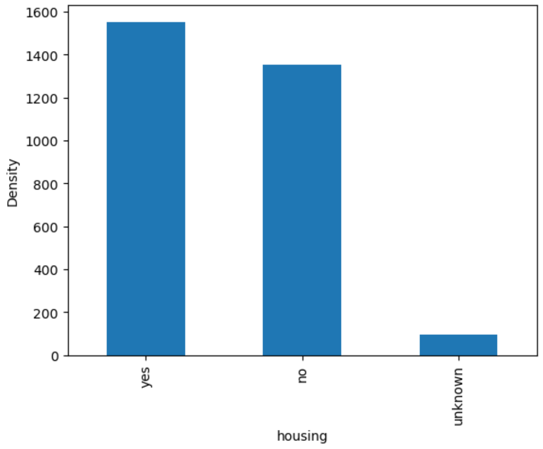


Figure 9. Default Pie chart

The Histogram above (Figure 8) indicates the values of attribute default with their frequency. Default attribute only has 2 values either it is a 'no' or 'unknown' (missing values). The mode in this dataset is 'no'. In contrast, the value 'unknown' has the smallest number of frequencies of only 630. The pie chart (Figure 9) is mostly comprised of 'no' which made up to 79% of the pie chart. In addition, we can also see that there's a large number of missing values about 21% identified as 'unknown' in this attribute.

Attribute Name: housing

Value	Frequency
yes	1552
no	1354
unknown	94



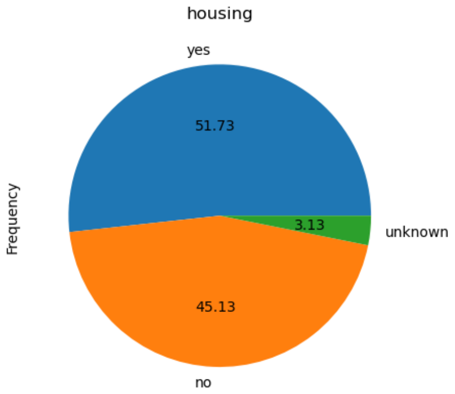


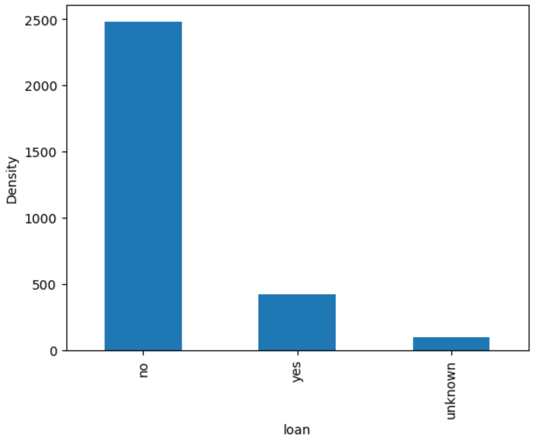
Figure 10. Housing Histogram

Figure 11. Housing Pie chart

The Histogram above (Figure 10) indicates the values of attribute housing with their frequency. In the given information it was outlined that housing refers to the housing loan of the people in this dataset. Utilizing that logic in this data, the mode in this dataset is ‘yes’ which has a frequency of 1552 which means most people in this dataset have a housing loan. In contrast, the value ‘no’ has the smallest number of frequencies of only 1354. The pie chart (Figure 11) is mostly comprised of ‘yes’ which made up to 51.73% of the pie chart. In addition, we can also see that there’s a small number of missing values about 3.13% identified as ‘unknown’ in this attribute.

Attribute Name: loan

Value	Frequency
no	2482
yes	424
unknown	94



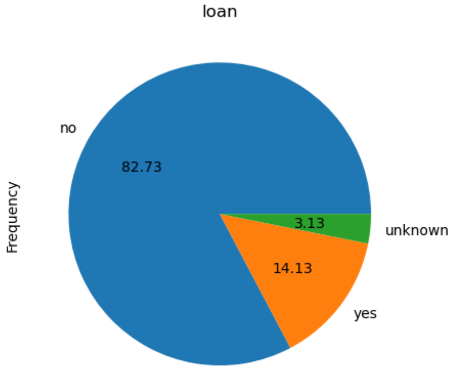


Figure 12. Loan Histogram

Figure 13. Loan Pie chart

The Histogram above (Figure 12) indicates the values of attribute loan with their frequency. In the given information it was outlined that loan refers to the personal loan of the people in this dataset. Utilizing that logic in this data, the mode in this dataset is ‘no’ which has a frequency of 2482 which mean most people in this dataset does not have a personal loan. In contrast, the value ‘yes’ has the smallest number of frequencies of only 424. The pie chart (Figure 13) is mostly comprised of ‘no’ which made up to 82% of the pie chart. In addition, we can also see that there’s a small number of missing values about 3.13% identified as ‘unknown’ in this attribute.

Attribute Name: contact

Value	Frequency
cellular	1868
telephone	1132

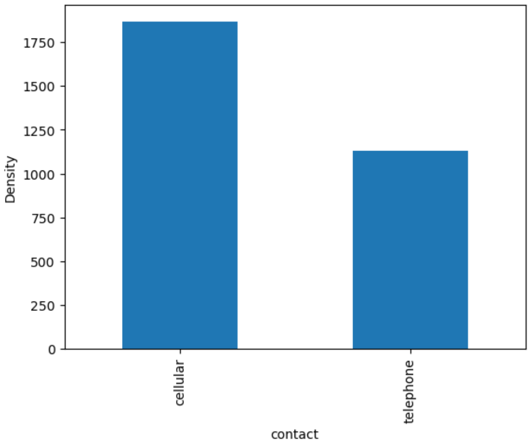


Figure 14. Contact Histogram

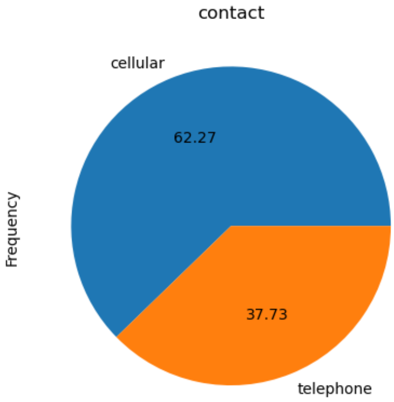


Figure 15. Contact Pie chart

The Histogram above (Figure 14) indicates the values of attribute contact with their frequency. contact attribute only has 2 values either it is a ‘cellular’ or ‘telephone’. In the given information it was outlined that contact refers to the contact communication type of the people in this dataset. Utilizing that logic in this data, the mode in this dataset is ‘cellular’. In contrast, the value ‘telephone’ has the smallest number of frequencies of only 1132. The pie chart (Figure 15) is mostly comprised of ‘cellular’ which made up to 62% of the pie chart.

Attribute Name: month

Value	Frequency
may	1048
jul	530
aug	387
jun	373
nov	312
apr	210
oct	47
mar	39
sep	35
dec	19

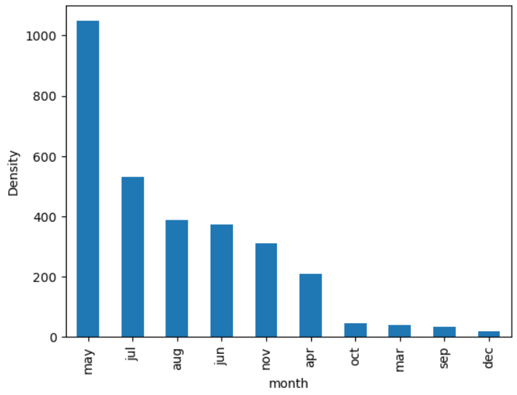


Figure 16. Month Histogram

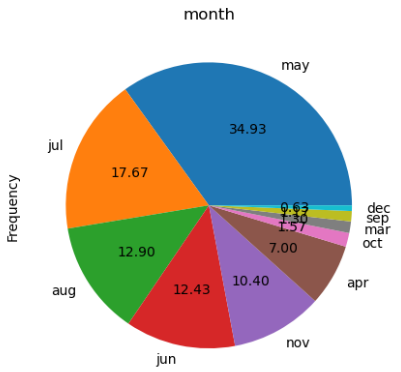


Figure 17. Month Pie chart

The Histogram above (Figure 16) indicates the last contact month of year with their frequency. The mode in this dataset is ‘may’ and followed by ‘july’ as the second highest. In contrast, ‘december’ has the smallest number of frequencies of only 19. This information can also be seen in the pie chart (Figure 17) where 34.93% of the whole population is last contacted on ‘may’ while only 0.63% is last contacted on ‘dec’. The pie chart is mostly comprised of ‘may’ and ‘july’ which made up to 52% of the pie chart.

Attribute Name: day_of_week

Value	Frequency
mon	669
thu	640
tue	599
wed	548
fri	544

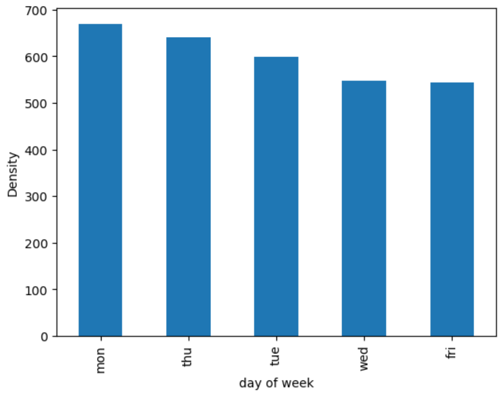


Figure 18. day of week Histogram

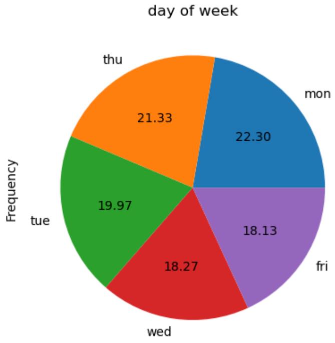


Figure 19. day of week Pie chart

The Histogram above (Figure 18) indicates the last contact day of week with their frequency. The mode in this dataset is ‘monday’ and followed by ‘thursday’ as the second highest. In contrast, ‘dec’ has the smallest number of frequencies of 544. Based on Figure 19, the values of day_of_week are evenly distributed as they roughly almost share the same weight in the pie chart.

Attribute Name: duration

Statistics	Value
Mean	264.914
Q1	102
Q2 (Median)	183
Q3	331
IQR	229
Maximum Value	3785
Minimum Value	3
Standard Deviation	270.948758

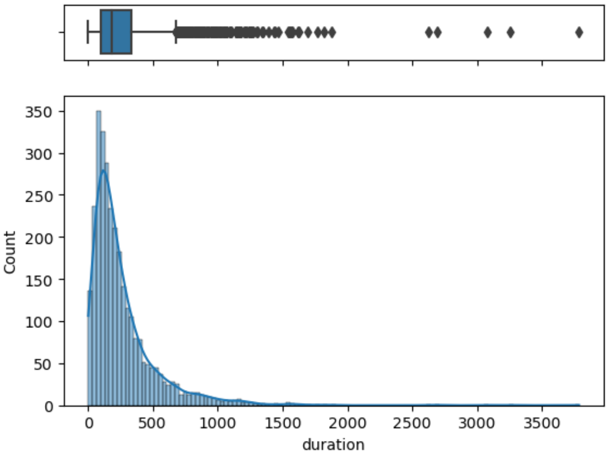


Figure 20. Duration Box plot and Histogram

In Figure 20, the graphs indicate the distribution of duration values in datasets. In the given information it was outlined that duration refers to the last contact duration in this dataset. Utilizing that logic in this data, the dataset has a range from 3 (Minimum value) to 3785 (maximum value). Through the visualization of box plot, duration attribute value is mainly concentrated around 102 to 331. This can also be seen in the histogram and the distribution lines which illustrate duration values majorly lie in the left side of the histogram roughly below the duration value of 500. In the Histogram, the curve of the distribution line is thin which implies that the dataset has low variations and data are less spread out.

Attribute Name: campaign

Statistics	Value
Mean	2.494
Q1	1
Q2 (Median)	2
Q3	3
IQR	2
Maximum Value	56
Minimum Value	1
Standard Deviation	2.78

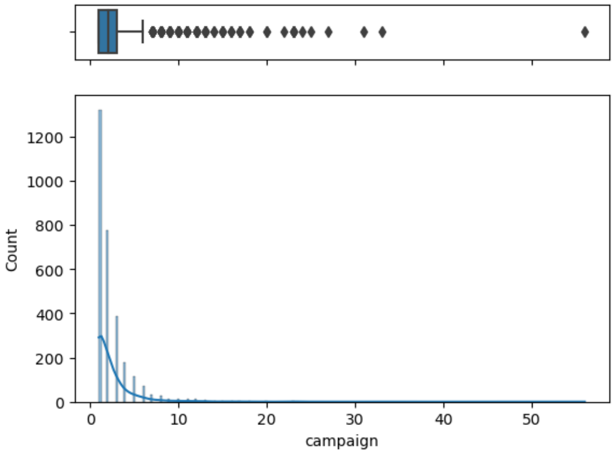


Figure 21. Campaign Box Plot and Histogram

In Figure 21, the graphs indicate the distribution of campaign values in datasets. In the given information it was outlined that campaign refers to the number of contacts performed during campaign in this dataset. Utilizing that logic in this data, the dataset has a range from 1 (minimum value) to 56 (maximum value). Through the visualization of box plot, campaign attribute value is mainly concentrated around 1 to 3. This can also be seen in the histogram and the distribution lines which illustrate campaign values majorly lie in the left side of the histogram. In the Histogram, the curve of the distribution line is thin which implies that the dataset has low variations and data are less spread out.

Attribute Name: pdays

Statistics	Value
Mean	964.902
Q1	999
Q2 (Median)	999
Q3	999
IQR	0
Maximum Value	999
Minimum Value	0
Standard Deviation	180.865

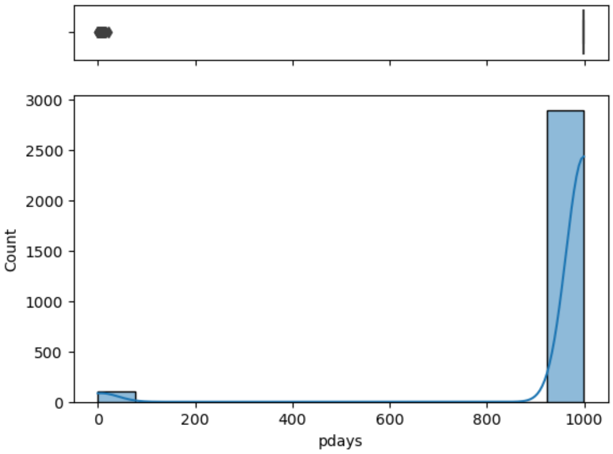


Figure 22. pdays Box Plot and Histogram

In Figure 22, the graphs indicate the distribution of pdays values in datasets. In the given information it was outlined that pdays refers to the number of days that passed by after the client was last contacted from a previous campaign. Utilizing that logic in this data, the dataset has a range from 0 (minimum value) to 999 (maximum value). Through the visualization of the box plot, pdays attribute value is mainly concentrated in the maximum value which is 999. This can also be seen in the histogram and the distribution lines which illustrate pdays values majorly lie in only the value 999. In the Histogram, the curve of the distribution line is thin which implies that the dataset has low variations and data are less spread out. Moreover, we can also see based on the box plot, the lower quartile is equal to the upper quartile, meaning the range of values are densely clustered in 1 value which is 999.

Attribute Name: previous

Statistics	Value
Mean	0.173
Q1	0
Q2 (Median)	0
Q3	0
IQR	0
Maximum Value	6
Minimum Value	0
Standard Deviation	0.49378

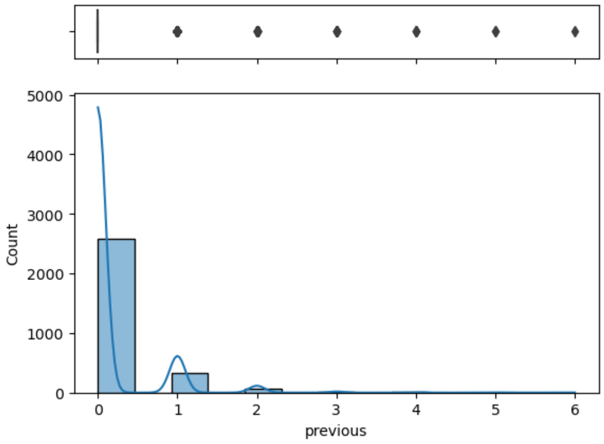
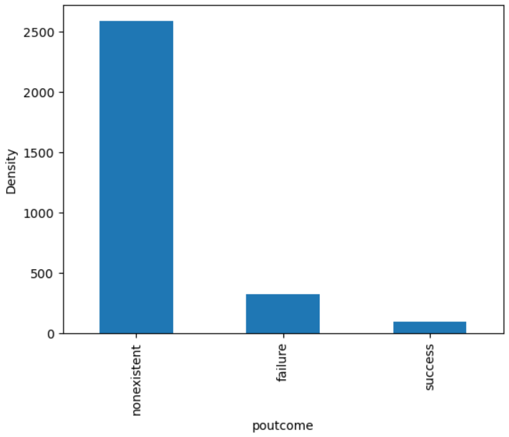


Figure 23. previous Box plot and Histogram

In Figure 23, the graphs indicate the distribution of previous values in datasets. In the given information it was outlined that previous refers to the number of contacts performed before the campaign. Utilizing that logic in this data, the dataset has a range from 0 (minimum value) to 6 (maximum value). Through the visualization of the box plot, previous attribute value is mainly concentrated in the minimum value which is 0. This can also be seen in the histogram and the distribution lines which illustrate previous values majorly lie in only the value 0. Moreover, we can also see based on the box plot, the lower quartile is equal to the upper quartile, meaning the range of values are densely clustered in 1 value which is 0.

Attribute Name: poutcome

Value	Frequency
nonexistent	2588
failure	320
success	92



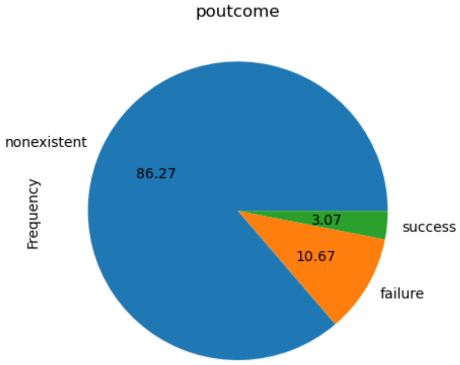


Figure 24. poutcome Histogram

Figure 25. poutcome Pie chart

The Histogram above (Figure 24) indicates the values of poutcome attribute with their frequency. In the given information it was outlined that poutcome refers to the outcome of the previous marketing campaign. Utilizing that logic in this data, the mode in this dataset is 'nonexistent' which has a frequency of 2588. In contrast, the value 'success' has the smallest number of frequencies of only 92. The pie chart (Figure 25) is mostly comprised of 'nonexistent' which made up to 82.27% of the pie chart.

Attribute Name: emp.var.rate

Statistics	Value
Mean	0.0494
Q1	-1.8
Q2 (Median)	1.1
Q3	1.4
IQR	3.2
Maximum Value	1.4
Minimum Value	-3.4
Standard Deviation	1.576

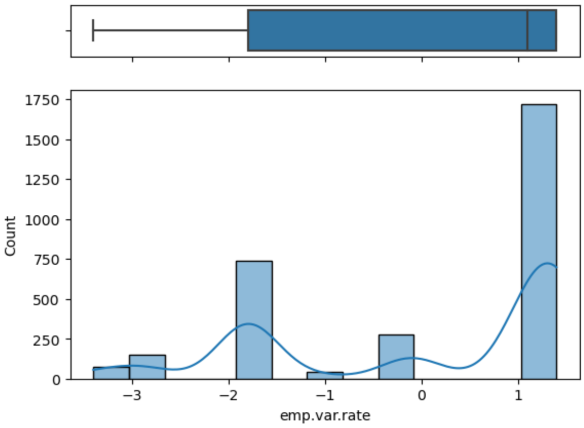


Figure 26. emp.var.rate Box plot and Histogram

In Figure 26, the graphs indicate the distribution of emp.var.rate values in datasets. In the given information it was outlined that emp.var.rate refers to the employment variation rate. Utilizing that logic in this data, the dataset has a range from -3.4 (minimum value) to 1.4 (maximum value). Through the visualization of the box plot, emp.var.rate attribute value is mainly concentrated in the maximum value which is 1.4. This can also be seen in the histogram and the distribution lines which illustrate emp.var.rate values majorly lie in only the value 1.4. In the Histogram, through the curve of the distribution line, we can see that the data are slightly spread out evenly. Moreover, we can also see based on the box plot, the range between upper quartile and lower quartile is large meaning it has a high variation.

Attribute Name: cons.price.idx

Statistics	Value
Mean	93.566
Q1	93.075
Q2 (Median)	93.798
Q3	93.994
IQR	0.919
Maximum Value	94.767
Minimum Value	92.2
Standard Deviation	0.582

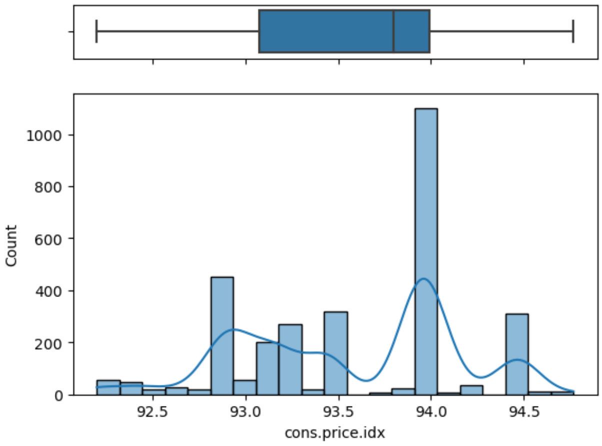


Figure 27. cons.price.idx Box plot and Histogram

In Figure 27, the graphs indicate the distribution of cons.price.idx values in datasets. In the given information it was outlined that cons.price.idx refers to consumer price index. Utilizing that logic in this data, the dataset has a range from 92.2 (Minimum value) to 94.767 (Maximum value). Through the visualization of the box plot, cons.price.idx attribute value is mainly concentrated in the value 93.994. This can also be seen in the histogram and the distribution lines which illustrate cons.price.idx values majorly lie in only the value 93.994. In the Histogram, through the curve of the distribution line, we can see that the data are approximately evenly spread out. Moreover, we can also see based on the box plot, the range between upper quartile and lower quartile is large meaning it has a high variation.

Attribute Name: cons.conf.idx

Statistics	Value
Mean	-40.62
Q1	-42.7
Q2 (Median)	-41.8
Q3	-36.4
IQR	6.3
Maximum Value	-26.9
Minimum Value	-50.8
Standard Deviation	4.65

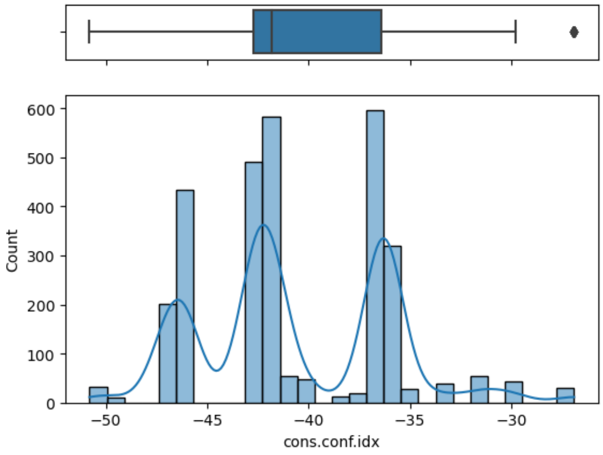


Figure 28. cons.conf.idx Box plot and Histogram

In Figure 28, the graphs indicate the distribution of cons.conf.idx values in datasets. In the given information it was outlined that cons.conf.idx refers to the consumer confidence index. Utilizing that logic in this data, the dataset has a range from -50.8 (Minimum value) to -26.9 (Maximum value). Through the visualization of the box plot, cons.conf.idx attribute value is mainly concentrated in 3 values (-36.4, -42.7, -46.2). This can also be seen in the histogram and the distribution lines which illustrate cons.conf.idx values majorly lie in only the -36.4. In the Histogram, through the curve of the distribution line, we can see that the data are evenly spread out. Moreover, we can also see based on the box plot, the range between upper quartile and lower quartile is large meaning it has a high variation.

Attribute Name: euribor3m

Statistics	Value
Mean	3.599
Q1	1.344
Q2 (Median)	4.857
Q3	4.961

IQR	3.617
Maximum Value	5.045
Minimum Value	0.636
Standard Deviation	1.732

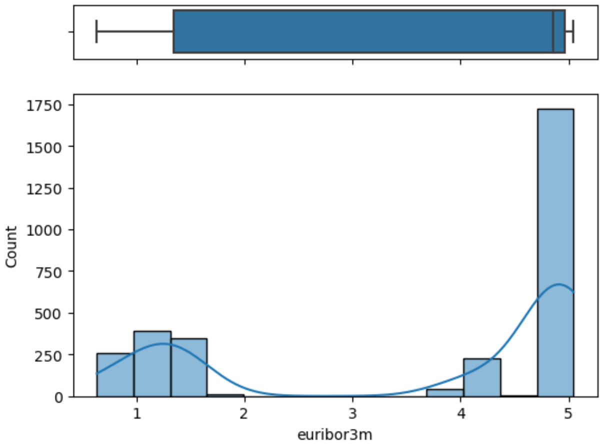


Figure 29. euribor3m Box plot and Histogram

In Figure 29, the graphs indicate the distribution of euribor3m values in datasets. In the given information it was outlined that euribor3m refers to the euribor 3 month rate. Utilizing that logic in this data, the dataset has a range from 0.636 (minimum value) to 5.045 (maximum value). Through the visualization of the box plot, euribor3m attribute value is concentrated on 2 sides (left and right). This can also be seen in the histogram and the distribution lines which illustrate euribor3m values majorly lie in value 5 and value 1. In the Histogram, through the curve of the distribution line, we can see that the data are evenly spread out. Moreover, we can also see based on the box plot, the range between upper quartile and lower quartile is large meaning it has a high variation.

Attribute Name: nr.employed

Statistics	Value
Mean	5166.54
Q1	5099
Q2 (Median)	5191
Q3	5228
IQR	129
Maximum Value	5228
Minimum Value	4963.6
Standard Deviation	70.55

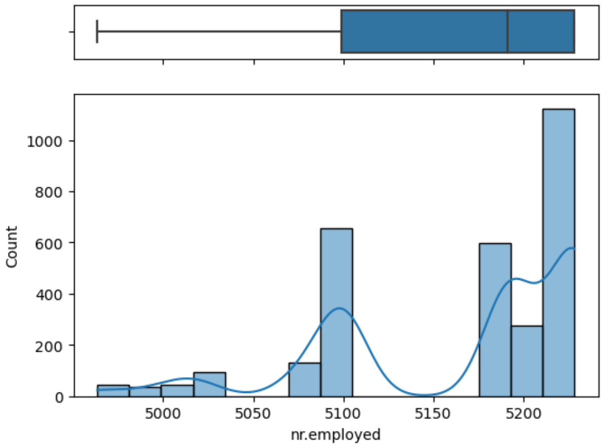
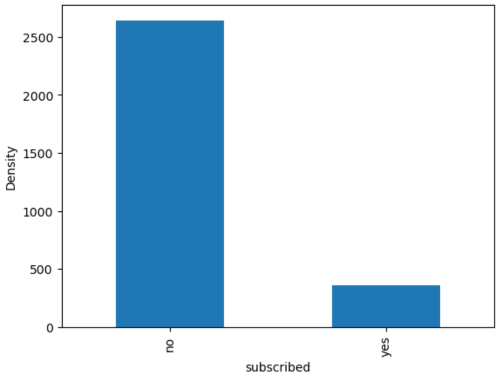


Figure 30. nr.employed Box plot and Histogram

In Figure 30, the graphs indicate the distribution of nr.employed values in datasets. In the given information it was outlined that nr.employed refers to the number of employees. Utilizing that logic in this data, the dataset has a range from 4963.6 (minimum value) to 5228 (maximum value). Through the visualization of the box plot, nr.employed attribute value is concentrated in the maximum value which is 5228.1. This can also be seen in the histogram and the distribution lines which illustrate nr.employed values majorly lie in value 5228.1 and value 5100. In the Histogram, through the curve of the distribution line, we can see that the data are evenly spread out. Moreover, we can also see based on the box plot, the range between upper quartile and lower quartile is large, implying that it has a high variation.

Attribute Name: subscribed

Value	Frequency
no	2642
yes	358



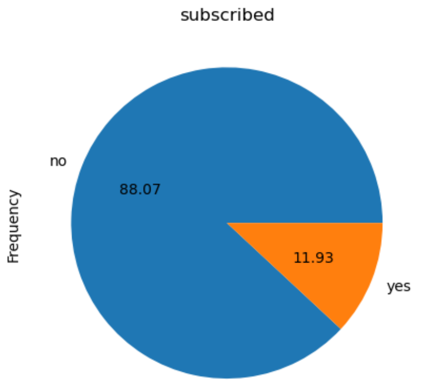


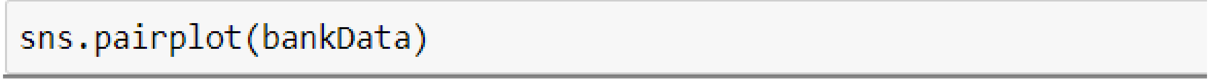
Figure 31. Subscribed Histogram

Figure 32. Subscribed Pie Chart

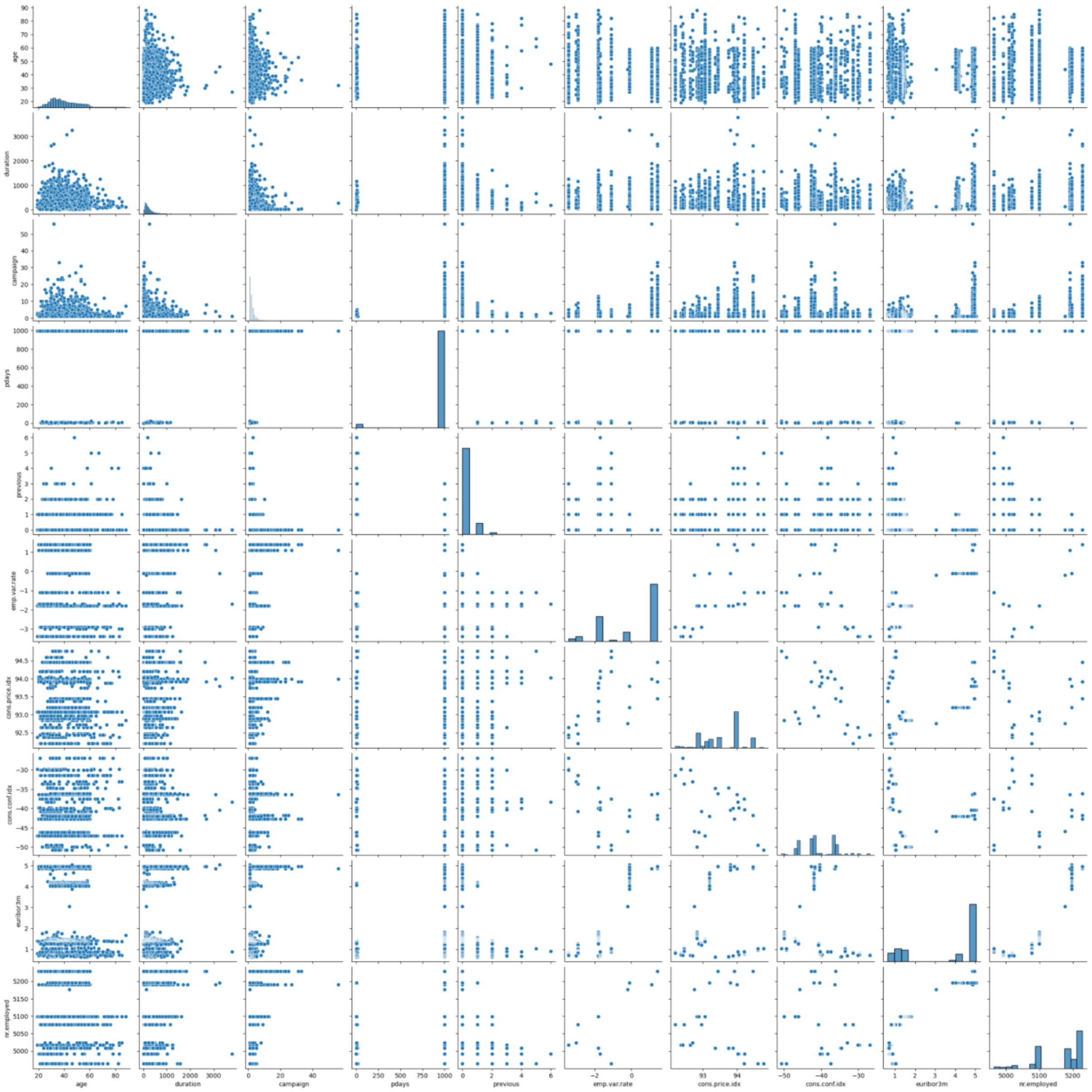
The Histogram above (Figure 31) indicates the value of subscribed. In the given information it was outlined that subscribed refers to whether a person subscribed to a term deposit. Utilizing that logic in this data, the mode or most of the people in this dataset do not subscribe to a term deposit. This information can also be seen in the pie chart (Figure 32) where 88.07% of the whole population does not subscribe while only 11.93% does subscribe.

3. Data Exploration

By using a clustering technique like K-means and agglomerative clustering for this data exploration section, we might be able to identify an interesting pattern and outlier in this data. To gain a bird eye view on our data, we can utilized this one line of python code:



```
sns.pairplot(bankData)
```



To see an attribute which has a strong correlation to another attribute, we can use pandas.DataFrame.corr() and then map it to heatmap to see the correlation.

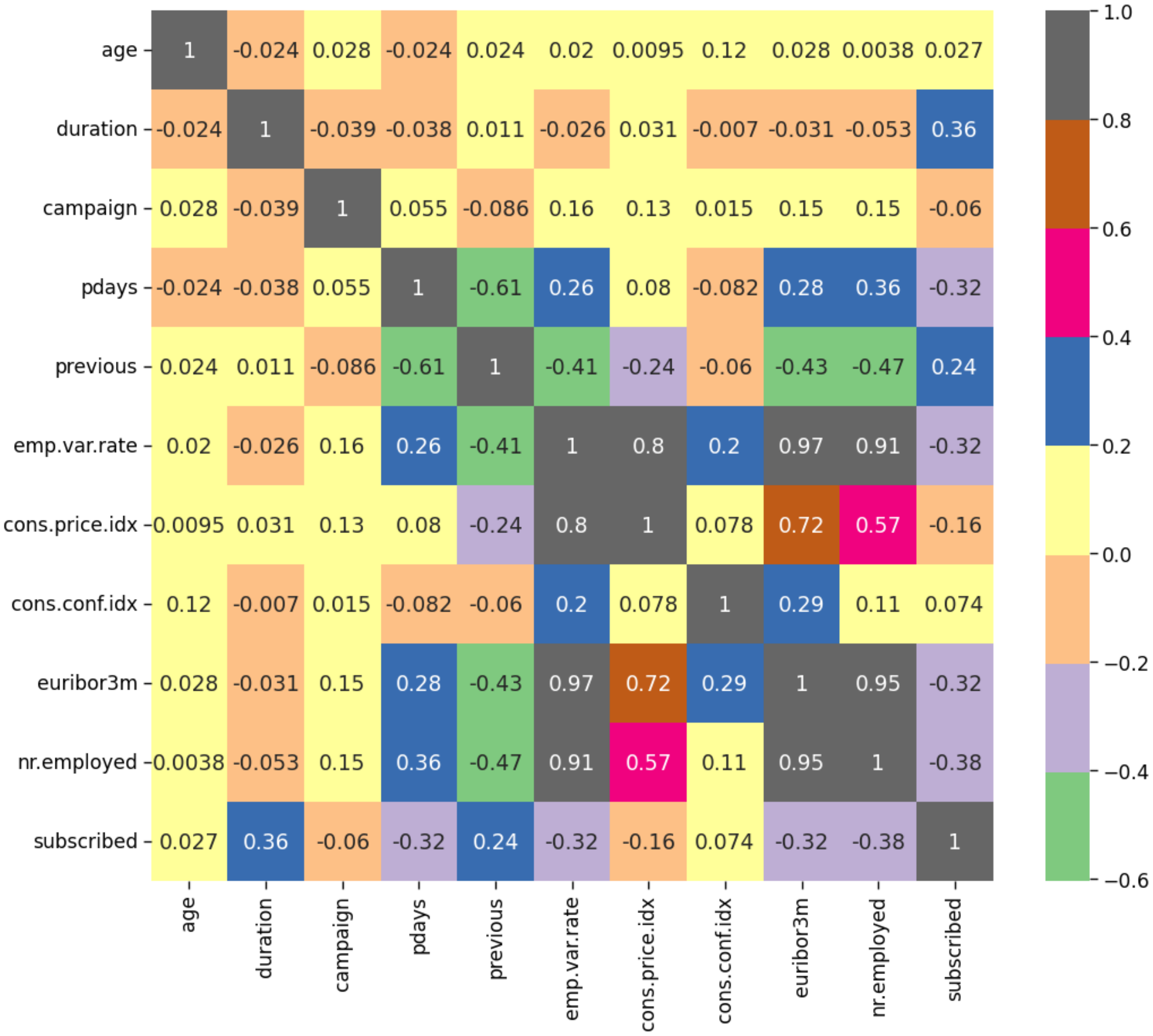
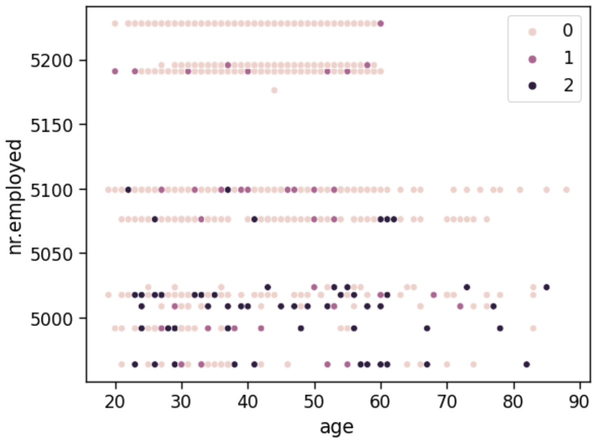


Figure 33. Heatmap of the dataset correlation of columns

Amongst all the features, several features like age vs nr.employed gives an interesting pattern and strong correlation. Thus, we will investigate it more using the clustering algorithm. By using the K-means algorithm, we can see that the K-means algorithm (figure 34) is able to group similar instances better than the agglomerative algorithm (figure 35). In figure 34, the upper part of the data points are grouped as the 0 cluster while in the middle part, the algorithm grouped the data points as the 1 cluster and the bottom part as the 2 cluster.



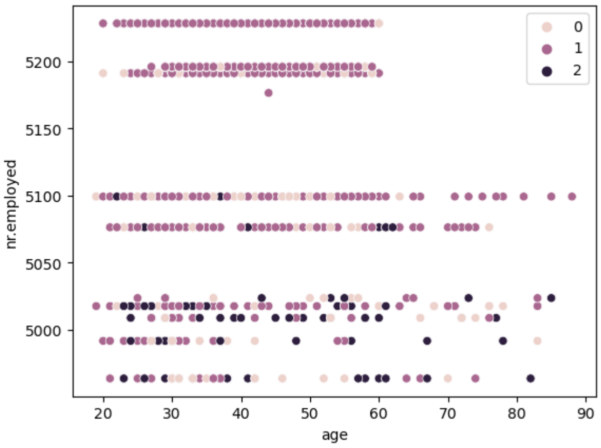


Figure 34. K-means age vs nr.employed

Figure 35. Agglomerative clustering age vs nr.employed

We can also see that there’s a few miss-identified classes in each cluster. This can be an interesting information that we can dive in and investigate further. Another column which has a strong correlation based on the heatmap is nr.employed and emp.var.rate. The result is shown below:

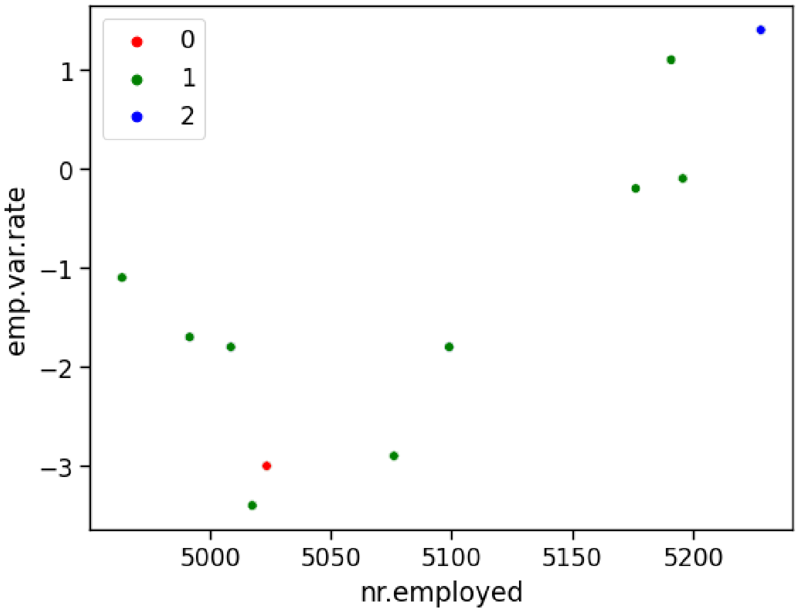


Figure 36. Agglomerative clustering age vs nr.employed

Based on the diagram above, we can see that the algorithm is able to distinguish 3 different clusters which are the right upper part (blue), the middle part (green), and the left bottom part (red). However, instead of seeing the correlation between numerical attributes, we can see the correlation between all the other attributes and the target attribute which is the subscribed attribute.

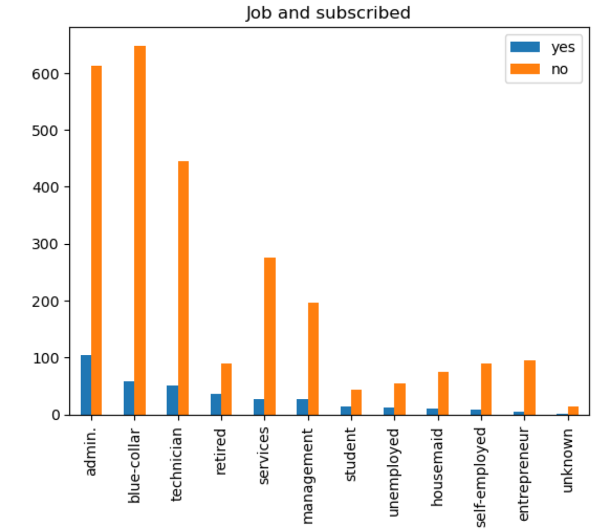


Figure 37. Bar Graph job vs subscribed

Based on the diagram above, we can see the number of people who subscribed to a term deposit based on their job. This will make our understanding of the data better as we can gain information and assumption of this data. Moreover, besides job and subscribed, we can also view the graph between marital and subscribed.

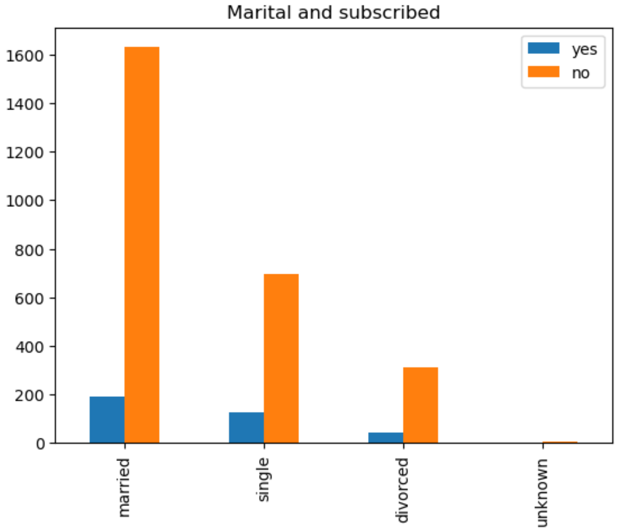


Figure 38. Bar Graph marital vs subscribed

Furthermore, to explore our dataset further we can also see the distribution of age by occupation. This will be useful to have a general idea on viewing the age of the people based on their job in this dataset. We are also able to see the lowest and highest value of age in each job. The result is shown below:

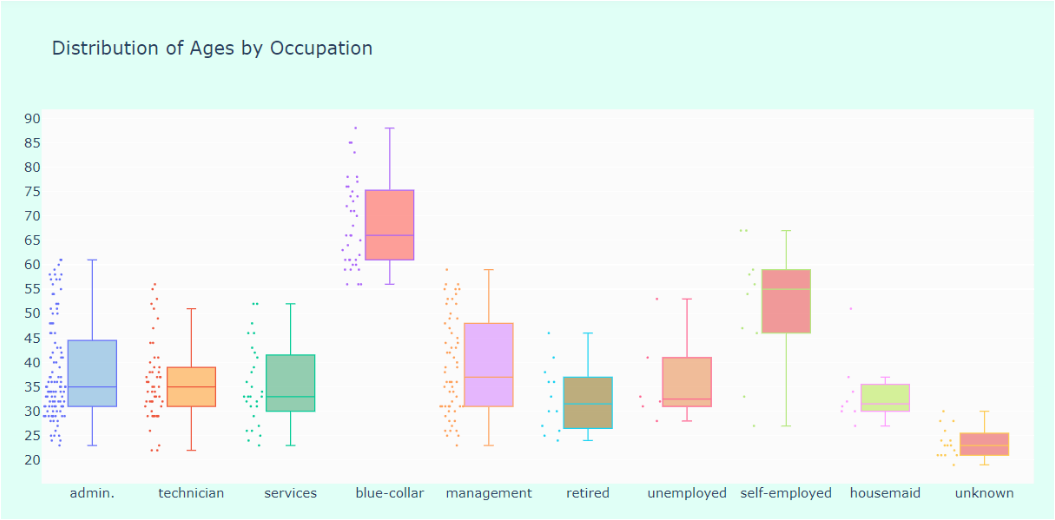
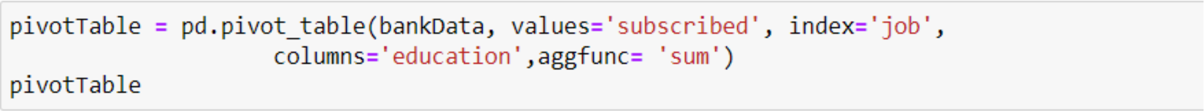
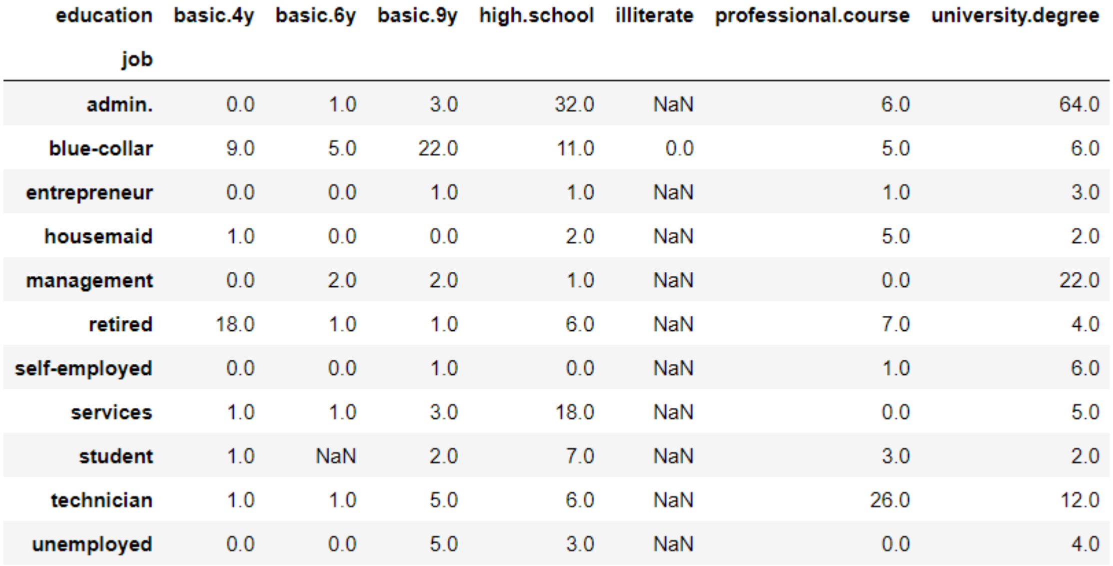


Figure 39. Distribution of Age by Job

Another method of exploration is using pivot table. We can use this line of code:



```
pivotTable = pd.pivot_table(bankData, values='subscribed', index='job',
                            columns='education',aggfunc= 'sum')
pivotTable
```



education	basic.4y	basic.6y	basic.9y	high.school	illiterate	professional.course	university.degree
job							
admin.	0.0	1.0	3.0	32.0	NaN	6.0	64.0
blue-collar	9.0	5.0	22.0	11.0	0.0	5.0	6.0
entrepreneur	0.0	0.0	1.0	1.0	NaN	1.0	3.0
housemaid	1.0	0.0	0.0	2.0	NaN	5.0	2.0
management	0.0	2.0	2.0	1.0	NaN	0.0	22.0
retired	18.0	1.0	1.0	6.0	NaN	7.0	4.0
self-employed	0.0	0.0	1.0	0.0	NaN	1.0	6.0
services	1.0	1.0	3.0	18.0	NaN	0.0	5.0
student	1.0	NaN	2.0	7.0	NaN	3.0	2.0
technician	1.0	1.0	5.0	6.0	NaN	26.0	12.0
unemployed	0.0	0.0	5.0	3.0	NaN	0.0	4.0

Figure 40. Pivot Table

Based on Figure 40, we can see the number of people based on their education and jobs. This will be useful for gaining an information and assumption based on these attributes. In the pivot table above, we can have a general idea on the distribution of job based on their education and we can derive an assumption from this pivot table.

1B. Data Pre-processing

Before we get into our data pre-processing part, we must first solve the problem of missing values in our dataset. In the Diagram (Figure 41) below, we can see where the missing values lie in our dataset.

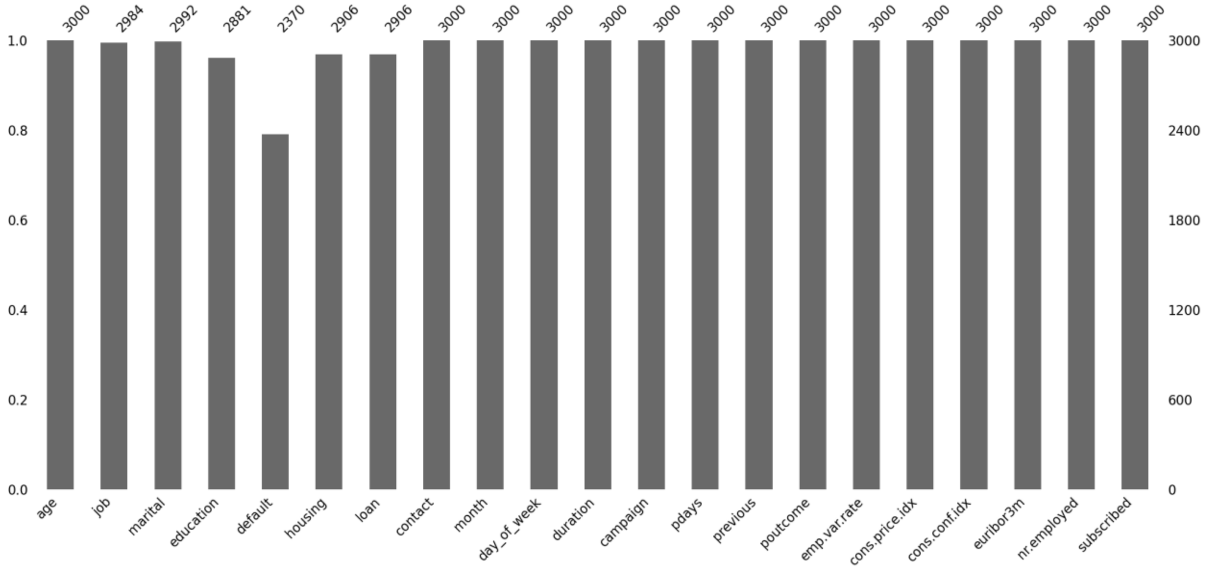
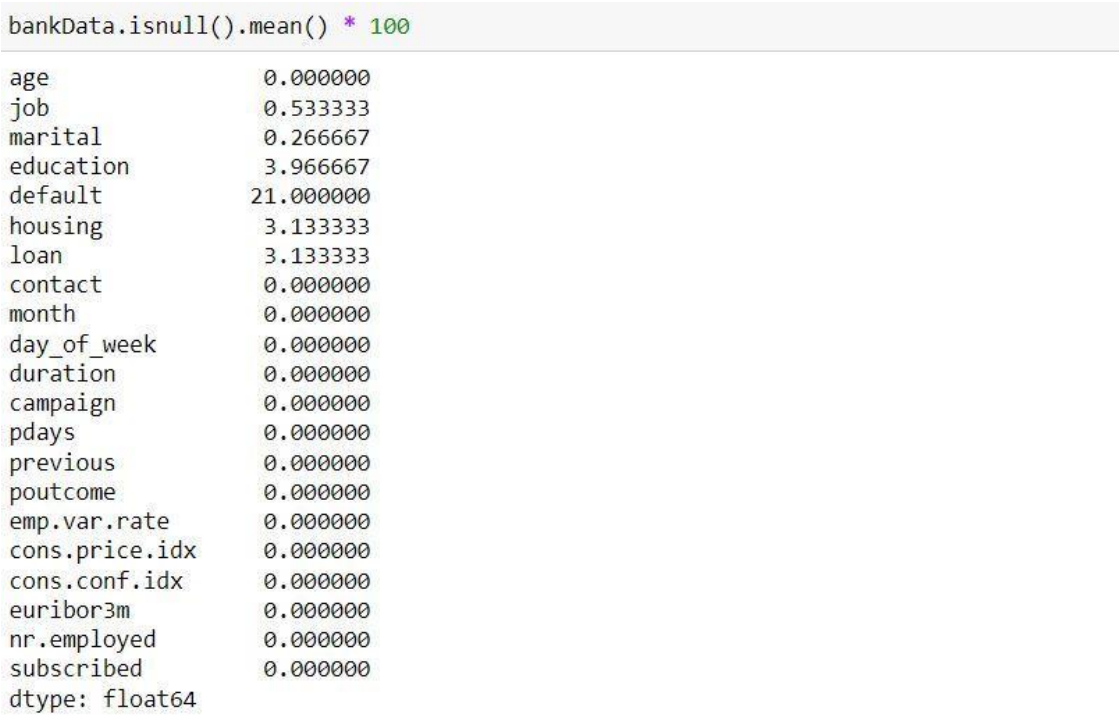


Figure 41. Missing value Diagram

To support our decision whether dropping or imputing those missing values, we must see the percentage of the missing values in each attribute and if it exceeds certain threshold (10%) then, that specific attribute should be drop instead of imputing their missing values. To do this we can execute 1 line of python code:

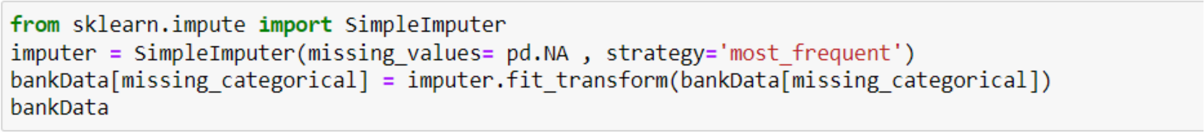


```
bankData.isnull().mean() * 100
```

```
age                0.000000
job                0.533333
marital            0.266667
education          3.966667
default           21.000000
housing            3.133333
loan               3.133333
contact            0.000000
month              0.000000
day_of_week        0.000000
duration           0.000000
campaign           0.000000
pdays              0.000000
previous           0.000000
poutcome           0.000000
emp.var.rate       0.000000
cons.price.idx     0.000000
cons.conf.idx      0.000000
euribor3m          0.000000
nr.employed        0.000000
subscribed         0.000000
dtype: float64
```

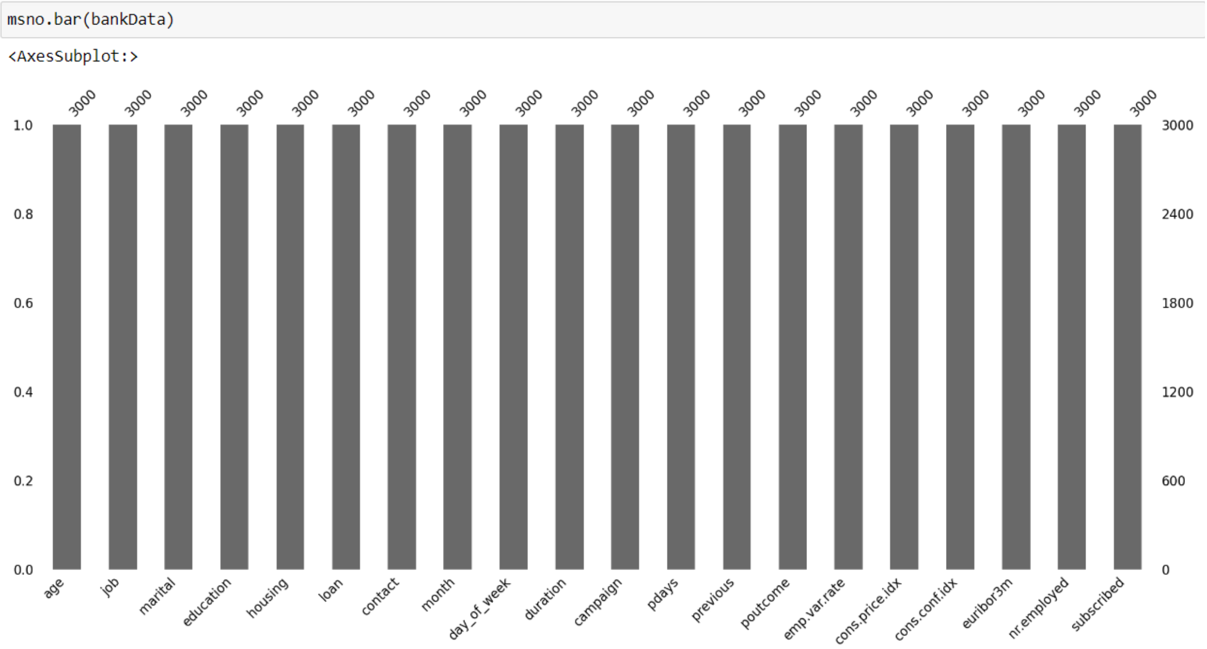
Figure 42. Missing value percentage

Based on Figure 42, the attribute default has a missing value of 21% which exceeds our threshold, thus it should be drop from our dataset as it may affect our data pre-processing and analysis. Whereas for the other missing values because they are categorical data, we can use a method called SimpleImputer (Figure ….) from scikit-learn library to impute it with the most frequent values (mode). After this imputing process, we must recheck our dataset again:



```python
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(missing_values= pd.NA , strategy='most_frequent')
bankData[missing_categorical] = imputer.fit_transform(bankData[missing_categorical])
bankData
```

Figure 43. Simple Imputer Python Code



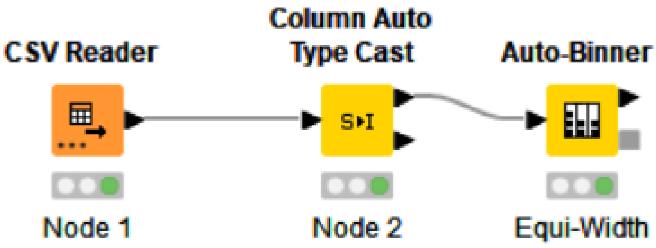
```
msno.bar(bankData)
```

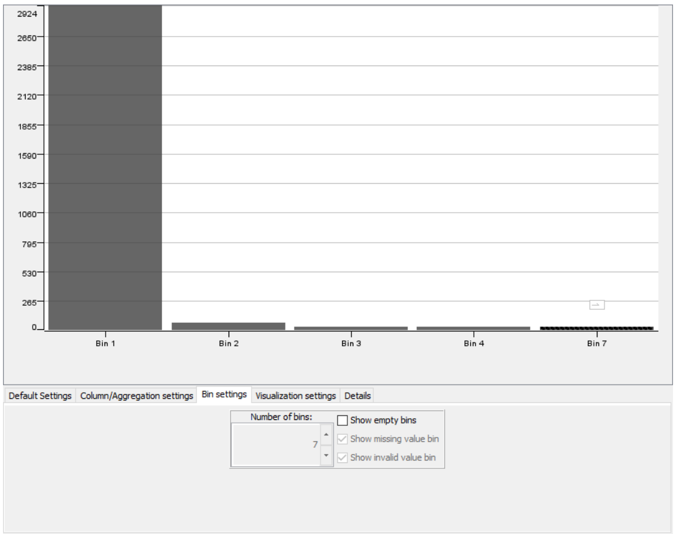
```
<AxesSubplot:>
```

Figure 44. Missing value Diagram

1.Binning Techniques

Equi-width binning





Number of bins: 7

□ Show empty bins

☑ Show missing value bin

☑ Show invalid value bin

Figure 45. Equi-Width on campaign

To do equi-width binning, we must first use node called "CSV Reader" to input our dataset and then connect it to a node called "Column Auto Type Cast" to identify the missing value in our dataset. Then, we proceed to connect it to a node called "Auto-Binner" to do binning. In this node configuration, we configure the attribute which we want to do binning and we must select the number of bins which is 7 and the type of binning we want to do which is equi-width. We select 7 as the number of bins because it gives an interesting pattern. Based on figure 45, when the bin number increase, the frequency decreases.

Bin 1: (1,9], Frequency: 2924

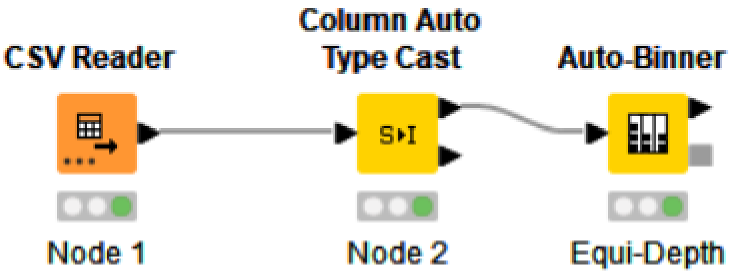
Bin 2: (9,17], Frequency: 60

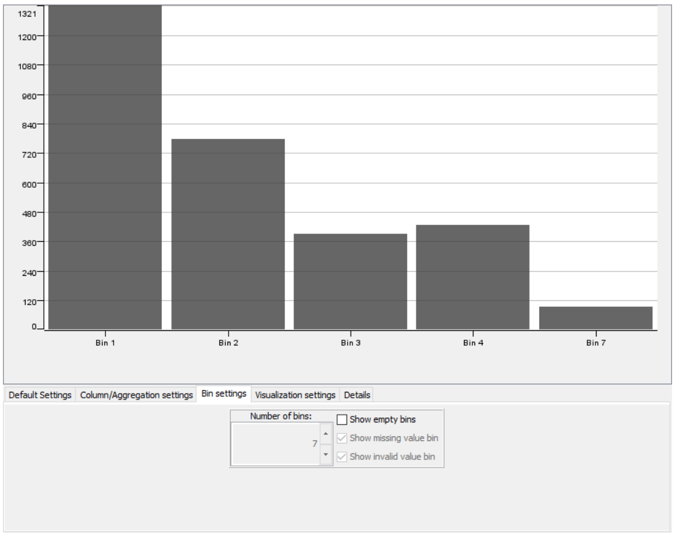
Bin 3: (17,25], Frequency: 12

Bin 4: (25,33], Frequency: 3

Bin 7: (49,56], Frequency: 1

Equi-depth binning





Number of bins: 7

□ Show empty bins

☑ Show missing value bin

☑ Show invalid value bin

Figure 46. Equi-Depth on campaign

To do equi-depth binning, we must first use node called “CSV Reader” to input our dataset and then connect it to a node called “Column Auto Type Cast” to identify the missing value in our dataset. Then, we proceed to connect it to a node called “Auto-Binner” to do binning. In this node configuration, we configure the attribute which we want to do binning and we must select the number of bins which is 7 and the type of binning we want to do which is equi-depth. We select 7 as the number of bins because it gives an interesting pattern. Based on figure 46, it gives us a pattern where when the bin number increase, the frequency decreases. But the rate of decrease is not drastic and it can be said that equi-depth with 7 number of bins are able to give even distribution.

Bin 1: (1,1], Frequency: 1321

Bin 2: (1,2], Frequency: 776

Bin 3: (2,3], Frequency: 387

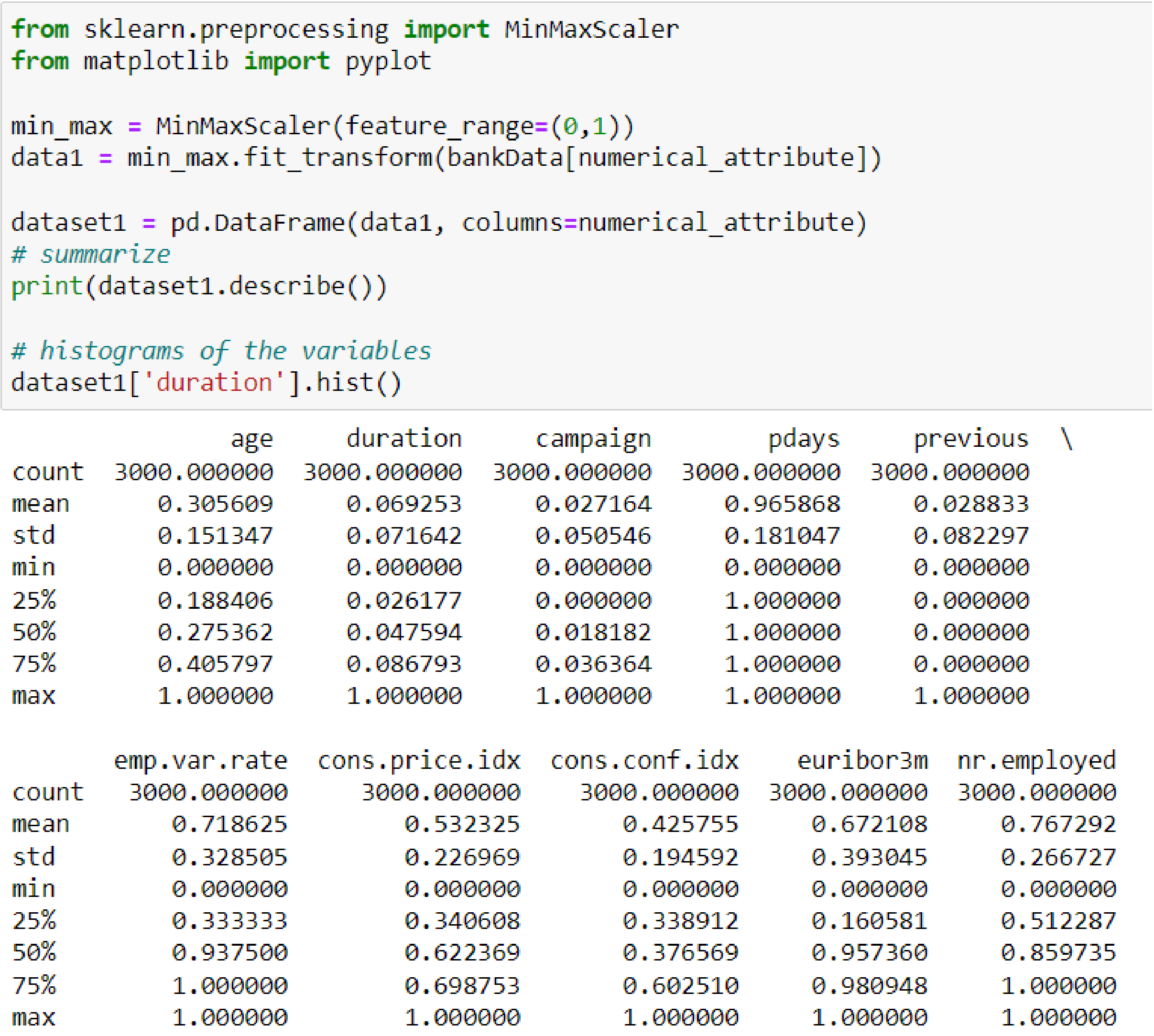
Bin 4: (3,8], Frequency: 425

Bin 7: (8,56], Frequency: 91

2.Normalization

The purpose of normalisation is to change the values of numeric columns in the dataset to use a common scale but preventing differences in the ranges of values or losing information. Moreover, we do normalization so that each feature can have the same level of importance as some Machine Learning algorithms tend to consider features with bigger range of number to be more important than the other. Normalization can only be applied for numerical data and not for categorical data.

Min-Max Normalization



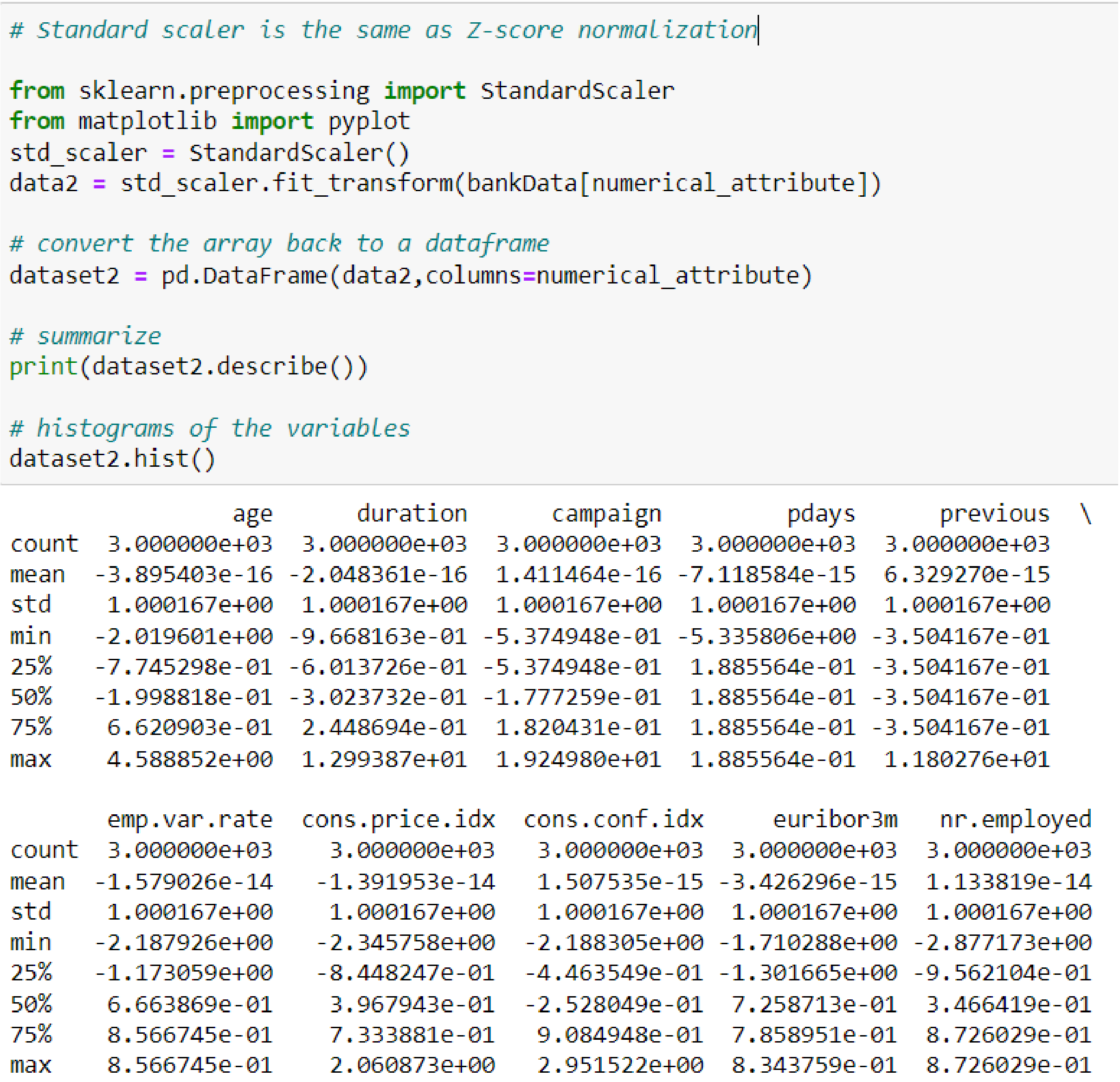
```python
from sklearn.preprocessing import MinMaxScaler
from matplotlib import pyplot

min_max = MinMaxScaler(feature_range=(0,1))
data1 = min_max.fit_transform(bankData[numerical_attribute])

dataset1 = pd.DataFrame(data1, columns=numerical_attribute)
# summarize
print(dataset1.describe())

# histograms of the variables
dataset1['duration'].hist()
```

```
               age     duration     campaign        pdays     previous  \
count  3000.000000  3000.000000  3000.000000  3000.000000  3000.000000   
mean      0.305609     0.069253     0.027164     0.965868     0.028833   
std       0.151347     0.071642     0.050546     0.181047     0.082297   
min       0.000000     0.000000     0.000000     0.000000     0.000000   
25%       0.188406     0.026177     0.000000     1.000000     0.000000   
50%       0.275362     0.047594     0.018182     1.000000     0.000000   
75%       0.405797     0.086793     0.036364     1.000000     0.000000   
max       1.000000     1.000000     1.000000     1.000000     1.000000   

       emp.var.rate  cons.price.idx  cons.conf.idx    euribor3m  nr.employed  
count   3000.000000     3000.000000    3000.000000  3000.000000  3000.000000  
mean       0.718625        0.532325       0.425755     0.672108     0.767292  
std        0.328505        0.226969       0.194592     0.393045     0.266727  
min        0.000000        0.000000       0.000000     0.000000     0.000000  
25%        0.333333        0.340608       0.338912     0.160581     0.512287  
50%        0.937500        0.622369       0.376569     0.957360     0.859735  
75%        1.000000        0.698753       0.602510     0.980948     1.000000  
max        1.000000        1.000000       1.000000     1.000000     1.000000  
```

Figure 47. Min-Max Normalization

With Min-Max normalization, we are able to determine the lowest and the highest value we want our attribute to be normalized. In this report, we choose 0 and 1 as the lowest and highest respectively. To do the Min-Max normalization, we use Python MinMaxScaler from scikit-learn library. The first step is to declare an instance of the MinMaxScaler class and then pass our dataset to the instance that we have just created. Lastly, we need to print out our dataset which has just been normalized.

Z-score Normalization



```
# Standard scaler is the same as Z-score normalization

from sklearn.preprocessing import StandardScaler
from matplotlib import pyplot
std_scaler = StandardScaler()
data2 = std_scaler.fit_transform(bankData[numerical_attribute])

# convert the array back to a dataframe
dataset2 = pd.DataFrame(data2,columns=numerical_attribute)

# summarize
print(dataset2.describe())

# histograms of the variables
dataset2.hist()
```

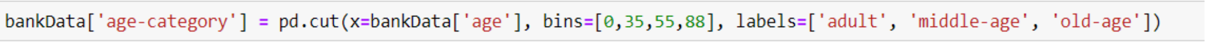
```
                age      duration      campaign         pdays      previous  \
count  3.000000e+03  3.000000e+03  3.000000e+03  3.000000e+03  3.000000e+03
mean  -3.895403e-16 -2.048361e-16  1.411464e-16 -7.118584e-15  6.329270e-15
std    1.000167e+00  1.000167e+00  1.000167e+00  1.000167e+00  1.000167e+00
min   -2.019601e+00 -9.668163e-01 -5.374948e-01 -5.335806e+00 -3.504167e-01
25%   -7.745298e-01 -6.013726e-01 -5.374948e-01  1.885564e-01 -3.504167e-01
50%   -1.998818e-01 -3.023732e-01 -1.777259e-01  1.885564e-01 -3.504167e-01
75%    6.620903e-01  2.448694e-01  1.820431e-01  1.885564e-01 -3.504167e-01
max    4.588852e+00  1.299387e+01  1.924980e+01  1.885564e-01  1.180276e+01

       emp.var.rate  cons.price.idx  cons.conf.idx     euribor3m   nr.employed
count  3.000000e+03    3.000000e+03   3.000000e+03  3.000000e+03  3.000000e+03
mean  -1.579026e-14   -1.391953e-14   1.507535e-15 -3.426296e-15  1.133819e-14
std    1.000167e+00    1.000167e+00   1.000167e+00  1.000167e+00  1.000167e+00
min   -2.187926e+00   -2.345758e+00  -2.188305e+00 -1.710288e+00 -2.877173e+00
25%   -1.173059e+00   -8.448247e-01  -4.463549e-01 -1.301665e+00 -9.562104e-01
50%    6.663869e-01    3.967943e-01  -2.528049e-01  7.258713e-01  3.466419e-01
75%    8.566745e-01    7.333881e-01   9.084948e-01  7.858951e-01  8.726029e-01
max    8.566745e-01    2.060873e+00   2.951522e+00  8.343759e-01  8.726029e-01
```

Figure 48. Z-score Normalization

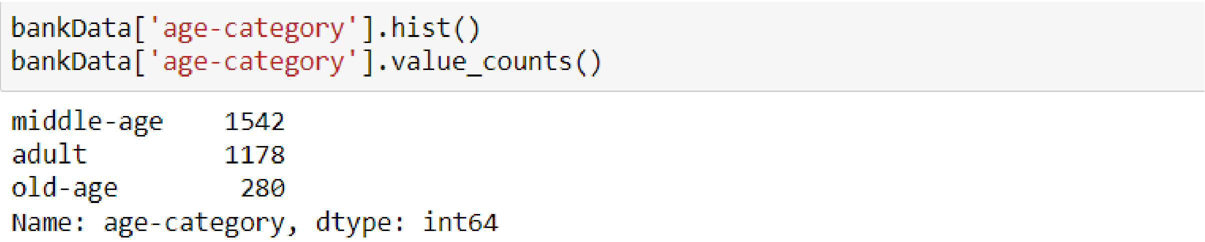
Z-score Normalization follows Standard Normal Distribution (SND). Therefore, it makes the mean equal to zero and scales the data to unit variance. In machine learning, Z-score is called Standard Scaler. To do the Z-score normalization, first we import the StandardScaler class from scikit-learn and we create an instance from that class. After creating an instance, we then pass our dataset to that instance and print out the result of the normalization.

3. Discretization

Discretization is the process of converting a numerical attribute into a categorical attribute more specifically ordinal attribute. In this case, age attribute will be discretized into 3 categories: adult, middle-age, and old-age. People whose age are below 35 will be categorized into adult category and those whose age are between 36 and 55 are categorized into middle-age category. Whereas people who are in older than 56 will be categorized as old-age. To do discretization, few lines of python is needed. The following code is shown below:



```
bankData['age-category'] = pd.cut(x=bankData['age'], bins=[0,35,55,88], labels=['adult', 'middle-age', 'old-age'])
```



```
bankData['age-category'].hist()
bankData['age-category'].value_counts()
```

```
middle-age    1542
adult         1178
old-age        280
Name: age-category, dtype: int64
```

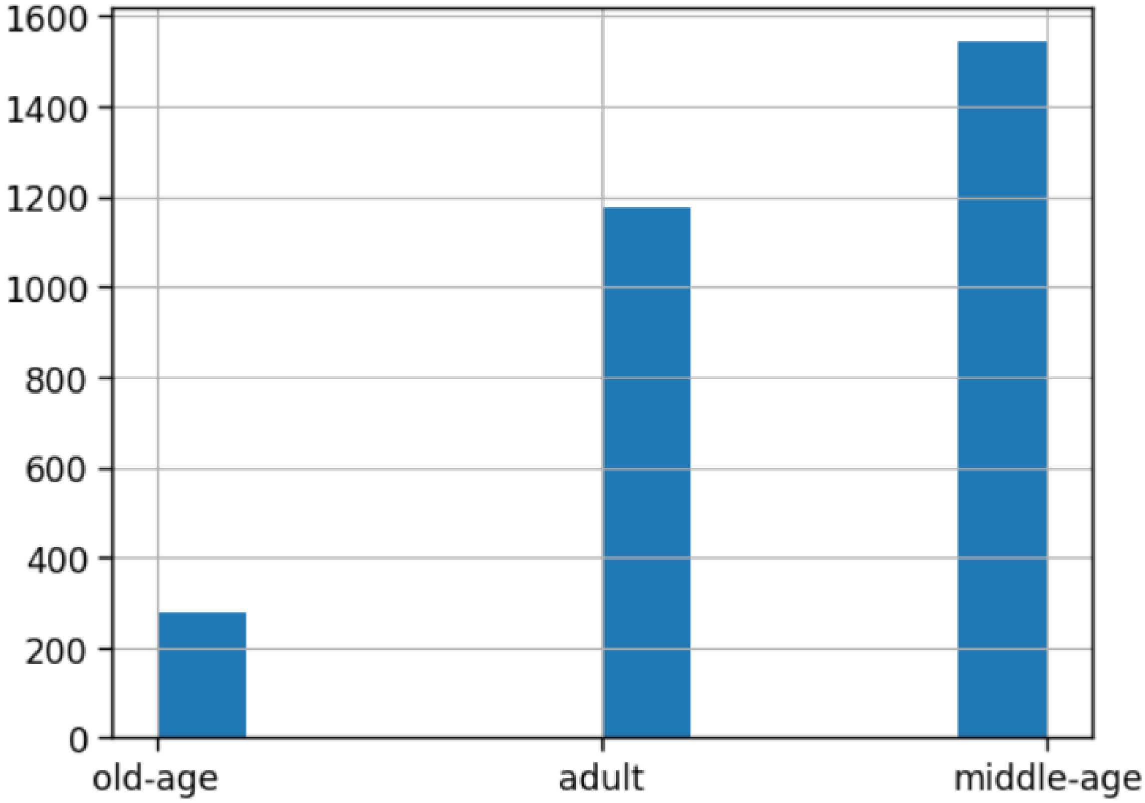
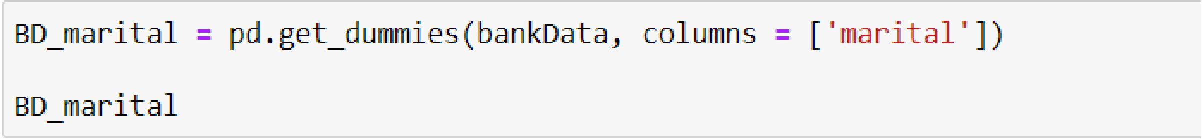


Figure 49. Discretization of age attribute

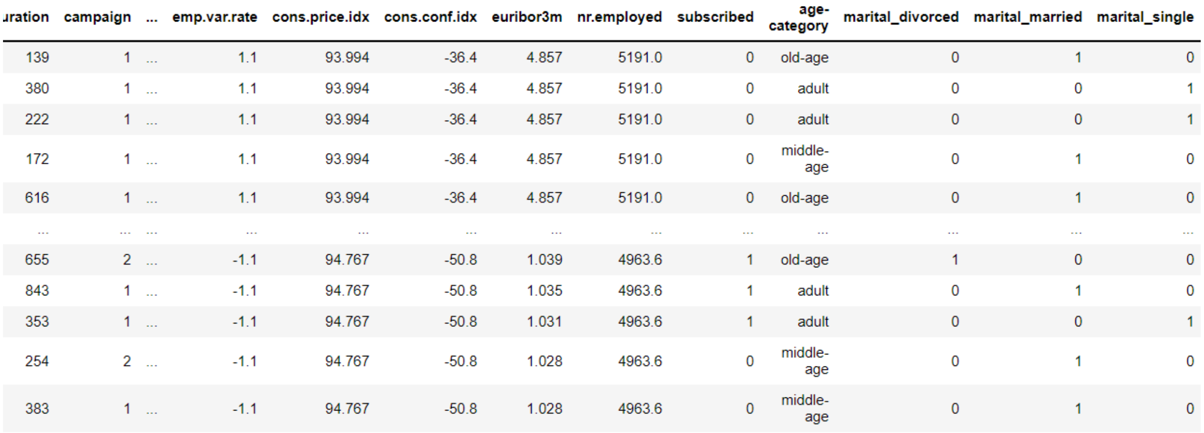
Based on the picture above, we can see that to do discretization in python, we just need to make a new column. The age attribute in the data frame is cut to segment and are sorted into bins based on several parameters. In the histogram above, middle-age has the highest frequency followed by adult who has the second highest frequency.

4. Binarization

Binarization is to map a continuous or categorical attribute into one or multiple binary variables (0 and 1). It is useful because in python, most algorithm are not able to process a categorical data. Therefore, by converting categorical attribute to numerical attribute, the feature can be included into the algorithm for further process like prediction. To do binarization in python, we can use this following code:



```
BD_marital = pd.get_dummies(bankData, columns = ['marital'])

BD_marital
```



uration	campaign	...	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed	subscribed	age-category	marital_divorced	marital_married	marital_single
139	1	...	1.1	93.994	-36.4	4.857	5191.0	0	old-age	0	1	0
380	1	...	1.1	93.994	-36.4	4.857	5191.0	0	adult	0	0	1
222	1	...	1.1	93.994	-36.4	4.857	5191.0	0	adult	0	0	1
172	1	...	1.1	93.994	-36.4	4.857	5191.0	0	middle-age	0	1	0
616	1	...	1.1	93.994	-36.4	4.857	5191.0	0	old-age	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...
655	2	...	-1.1	94.767	-50.8	1.039	4963.6	1	old-age	1	0	0
843	1	...	-1.1	94.767	-50.8	1.035	4963.6	1	adult	0	1	0
353	1	...	-1.1	94.767	-50.8	1.031	4963.6	1	adult	0	0	1
254	2	...	-1.1	94.767	-50.8	1.028	4963.6	0	middle-age	0	1	0
383	1	...	-1.1	94.767	-50.8	1.028	4963.6	0	middle-age	0	1	0

Figure 50. Binarization on marital attribute

Based on the picture above, we can see that to do binarization in python, we just need to use a function Pandas.get_dummies() where it will convert the categorical data we specify in the argument into a binary values as show above. For instance, a row who has marital status of 'married' will have a value 1 in the marital_married columns and will have value 0 in marital_divorced and marital_single.

1C. Summary

The most important findings of this report include the following:

- In education attribute, of all the values in that attribute, 'illiterate' only has 1 frequency which is significantly lower than all other value. This could indicate a noise or unique finding in education features as from the 3000 data, there's only 1 instance of it. There may be 2 reasons for this, it represents a data collection error rendering it as an outlier as it is possible that the data quality can be affected by human error when recording data.
- Default attribute has only 1 value which is 'no'. When we look at the statistical summary on default attribute, we can see that 79% of the whole instances has a 'no' value while 21% has a 'unknown' or missing value. This has few interesting findings for there's only 1 type of value in the attribute which we know and 21% of the 3000 instances has a missing value which exceed our

missing value threshold. Therefore, for these same reasons, default attribute is dropped.

- Unanswered and confidential – this could potentially explain the 21% of blank data/missing values in regard to their personal information. This is worth investigating further to check if the data has missing values that need to be recovered.
- Most of the numerical attributes have similar value where it can be seen on the distribution graph. The graph shows less variation as the data points tend to cluster around 1 or 2 values only.
- The attribute month does not have a year specified with it. This cause a several confusions as we can't compared which one is more recent than the other because there isn't any year specified. This is worth investigating further as month attribute which we can use as ordinal attribute would play an important role in understanding the data deeper.
- The attribute day_of_week has an equal distribution as shown in their histogram as each value have similar numbers of frequency. Thus, the data are spread out evenly for this attribute.
- Pdays attribute has a high frequency only in 1 attribute and the mean is in the mode and the maximum value. Furthermore, pdays attribute has 0 Interquartile range (IQR) as shown in its box plot. Hence, this attribute has less variation and the data are less spread out.
- By using the K-means algorithm, we can see that the K-means algorithm is able to group similar instances into 3 group for age vs nr.employed. They are slightly able to group similar instance and give interesting findings.
- In the bar graph between job and subscribed, we are able to see the people who subscribed to a term deposit based on their job. This is a huge finding as we can directly make an assumption and probability based on their jobs. We can then compute the probability of a person who subscribed to a term deposit for each job by diving the number of people who subscribed with the total number of people in that certain job.
- In the bar graph between marital and subscribed, we are able to see the people who subscribed to a term deposit based on their marital status. This is a huge finding as we can directly make an assumption and probability based on their status. We can then compute the probability of a person to subscribed to a term deposit for each marital status by diving the number of people who subscribed with the total number of people in that marital status.