131
```
# importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sbn
```

132
```
file_path = './data/watu_sales_data.csv'
dataset = pd.read_csv(file_path)
```

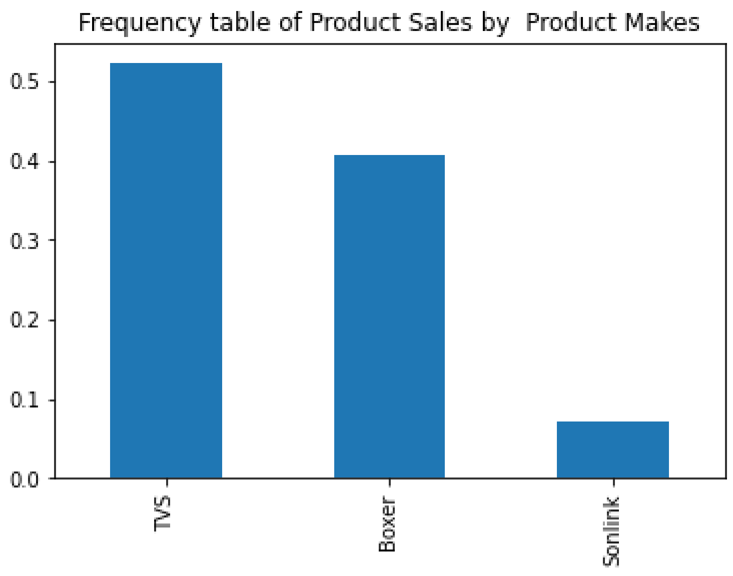
133
```
# exploration
print(dataset)
```

```
       make          model         make_model   branch date_of_sale     month  \
0     Boxer         BM 150       Boxer BM 150  Nairobi   06/01/2021   January
1       TVS      HLX 100CC      TVS HLX 100CC  Bungoma   22/04/2021     April
2       TVS      HLX 100CC      TVS HLX 100CC  Bungoma   26/02/2021  February
3       TVS      HLX 150CC      TVS HLX 150CC  Nairobi   26/03/2021     March
4       TVS        HLX 150        TVS HLX 150  Mombasa   05/06/2021      June
...     ...            ...                ...      ...          ...       ...
1995    TVS      HLX 100CC      TVS HLX 100CC  Bungoma   19/05/2021       May
1996    TVS  STAR 100CC KS  TVS STAR 100CC KS  Bungoma   18/03/2021     March
1997    TVS      HLX 150CC      TVS HLX 150CC  Nairobi   19/05/2021       May
1998    TVS        HLX 150        TVS HLX 150  Nairobi   15/01/2021   January
1999    TVS      HLX 125CC      TVS HLX 125CC   Migori   26/05/2021       May

      is_late_to_pay is_late_to_pay_word  payment_expected  payment_actual  \
0                  0                  NO             95394           95394
1                  0                  NO            102784          102784
2                  0                  NO             82295           82295
3                  0                  NO             93586           93586
4                  0                  NO            135065          135065
...              ...                 ...               ...             ...
1995               0                  NO            109048          109048
1996               0                  NO             93525           93525
1997               0                  NO            126081          126081
1998               0                  NO            105277          105277
1999               0                  NO            120462          120462

      payment_ratio  payment_full payment_full_word
0             100.0             0                NO
1             100.0             0                NO
2             100.0             0                NO
3             100.0             0                NO
4             100.0             0                NO
...             ...           ...               ...
1995          100.0             0                NO
1996          100.0             0                NO
1997          100.0             0                NO
1998          100.0             0                NO
1999          100.0             0                NO

[2000 rows x 13 columns]
```

134
```
# Understanding the Dataset
make_proportion = dataset['make'].value_counts(normalize=True)*100

print(make_proportion)

"""
Result:
The Brand Popularity is as follows
TVS         52.25
Boxer       40.70
```

```
Sonlink    7.05

TVS is the most popular make to sale the sales team my like to work on a partnership with TVS
to the data
"""
dataset['make'].value_counts(normalize=True).plot.bar(title='Frequency table of Product Sales
# plt.hist(make_proportion)
# plt.show()


TVS        52.25
Boxer      40.70
Sonlink     7.05
Name: make, dtype: float64
```

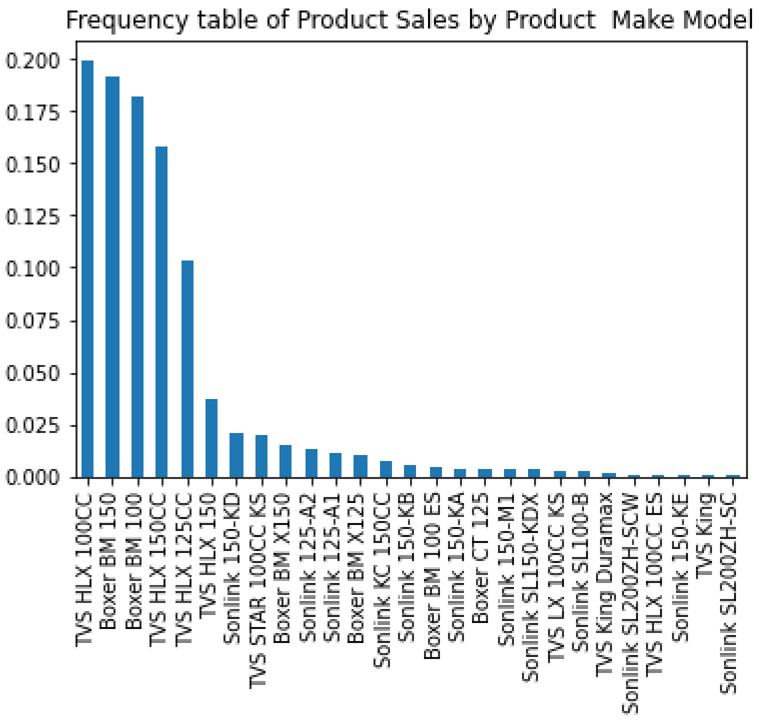
134
```
<AxesSubplot:title={'center':'Frequency table of Product Sales by  Product Makes'}>
```



135
```
model_proportion = dataset['make_model'].value_counts(normalize=True)*100
print(model_proportion)
"""
Results:
The model popularity is as follows:
BM 150                  19.20
BM 100                  18.20
HLX 100CC KS            12.70
HLX 125CC Refresh       10.30
HLX 150CC (5 Gears)     10.25
HLX 100CC ES PLUS        7.20
HLX 150CC ES             5.60
HLX 150X                 2.40
150-KD                   2.05
STAR 100CC KS            2.00
BM X150                  1.45
125-A2                   1.30
HLX 150X (5 Gears)       1.30
125-A1                   1.10
BM X125                  1.05
KC 150CC                 0.75
150-KB                   0.50
BM 100 ES                0.45
CT 125                   0.35
150-KA                   0.35
150-M1                   0.30
SL150-KDX                0.30
LX 100CC KS              0.25
SL100-B                  0.20
King Duramax             0.15
SL200ZH-SCW              0.10
HLX 100CC ES             0.05
```

```
150-KE                  0.05
King                    0.05
SL200ZH-SC              0.05
"""
dataset['make_model'].value_counts(normalize=True).plot.bar(title="Frequency table of Product

TVS HLX 100CC          19.90
Boxer BM 150           19.20
Boxer BM 100           18.20
TVS HLX 150CC          15.85
TVS HLX 125CC          10.30
TVS HLX 150             3.70
Sonlink 150-KD          2.05
TVS STAR 100CC KS       2.00
Boxer BM X150           1.45
Sonlink 125-A2          1.30
Sonlink 125-A1          1.10
Boxer BM X125           1.05
Sonlink KC 150CC        0.75
Sonlink 150-KB          0.50
Boxer BM 100 ES         0.45
Sonlink 150-KA          0.35
Boxer CT 125            0.35
Sonlink 150-M1          0.30
Sonlink SL150-KDX       0.30
TVS LX 100CC KS         0.25
Sonlink SL100-B         0.20
TVS King Duramax        0.15
Sonlink SL200ZH-SCW     0.10
TVS HLX 100CC ES        0.05
Sonlink 150-KE          0.05
TVS King                0.05
Sonlink SL200ZH-SC      0.05
Name: make_model, dtype: float64
```

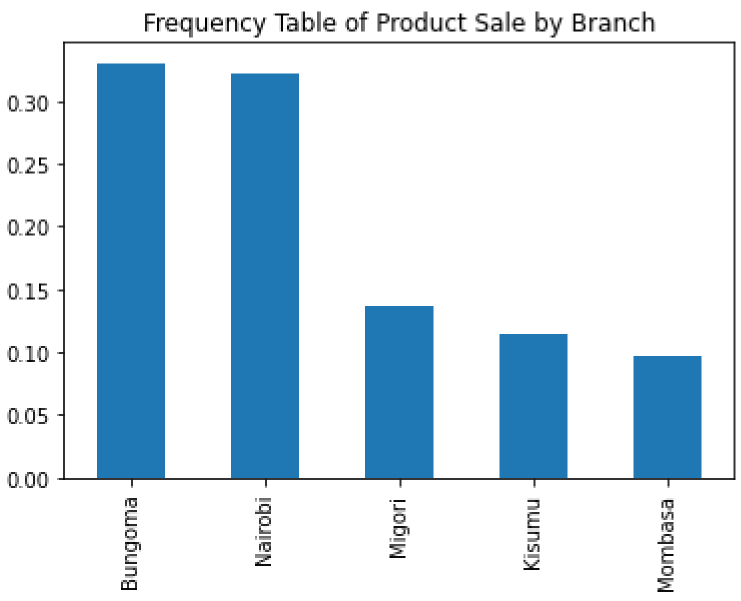
135
```
<AxesSubplot:title={'center':'Frequency table of Product Sales by Product  Make Model'}>
```



136
```
branch_proportion = dataset['branch'].value_counts(normalize=True)*100
print(branch_proportion)
"""
Bungoma is the most performing branch with 33.05% of all sale follow by
Nairobi at 32.20%, Migori at 13.65%, Kisumu at 11.45%, and Mombasa at 9.65% in the order
The sell team should investigate by two branches are making 65.25% of all the sale and what c
```

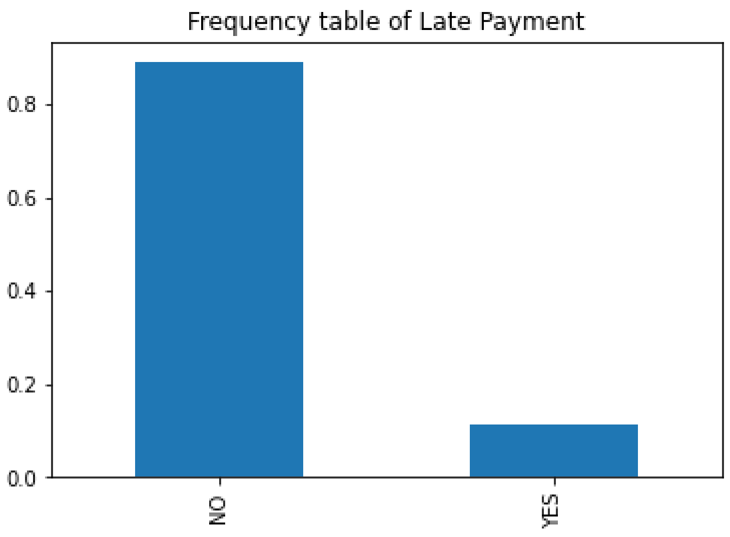
```
"""
dataset['branch'].value_counts(normalize=True).plot.bar(title='Frequency Table of Product Sal

Bungoma    33.05
Nairobi    32.20
Migori     13.65
Kisumu     11.45
Mombasa     9.65
Name: branch, dtype: float64
```

136
```
<AxesSubplot:title={'center':'Frequency Table of Product Sale by Branch'}>
```



137
```
late_payment_proportion = dataset['is_late_to_pay'].value_counts(normalize=True)*100
print(late_payment_proportion)

"""
11.1% of all loan applicant are late to make there repayments this ratio is bellow the kenya
to https://www.cgap.org/blog/its-time-slow-digital-credits-growth-east-africa
The credit department should make sure that this rate is reduced by offering financial advice
especially late borrowers
"""
dataset['is_late_to_pay_word'].value_counts(normalize=True).plot.bar(title="Frequency table o

0    88.9
1    11.1
Name: is_late_to_pay, dtype: float64
```

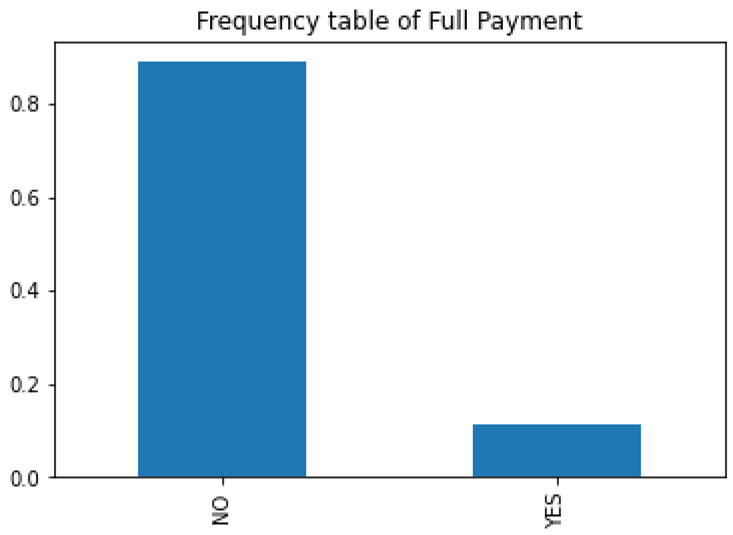
137
```
<AxesSubplot:title={'center':'Frequency table of Late Payment'}>
```



138
```
full_payment_proportion = dataset['payment_full'].value_counts(normalize=True)*100
print(full_payment_proportion)
"""
The credit team should also not that 11.1% of all repayments made where partial payment
"""
dataset['payment_full_word'].value_counts(normalize=True).plot.bar(title="Frequency table of
```

```
0    88.9
1    11.1
Name: payment_full, dtype: float64
```

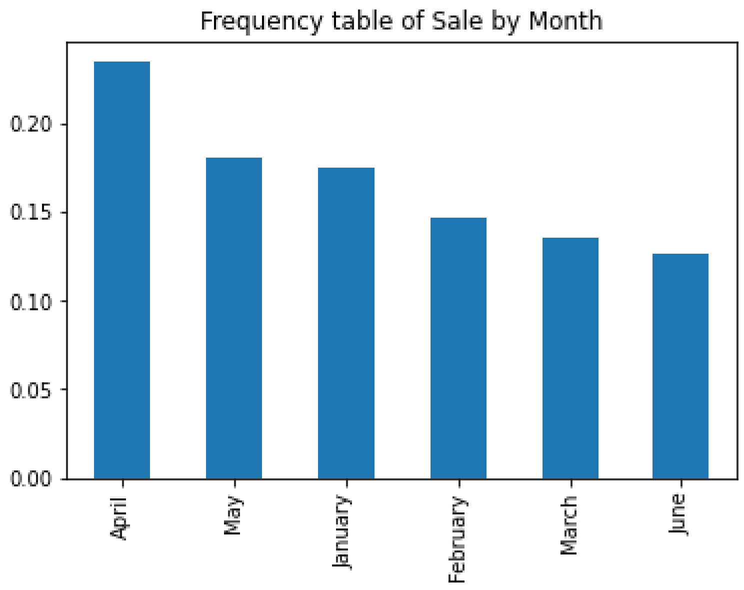
138
```
<AxesSubplot:title={'center':'Frequency table of Full Payment'}>
```



139
```
full_payment_proportion = dataset['month'].value_counts(normalize=True)*100
print(full_payment_proportion)
dataset['month'].value_counts(normalize=True).plot.bar(title="Frequency table of Sale by Mont
```

```
April       23.50
May         18.05
January     17.55
February    14.65
March       13.55
June        12.70
Name: month, dtype: float64
```

139
```
<AxesSubplot:title={'center':'Frequency table of Sale by Month'}>
```

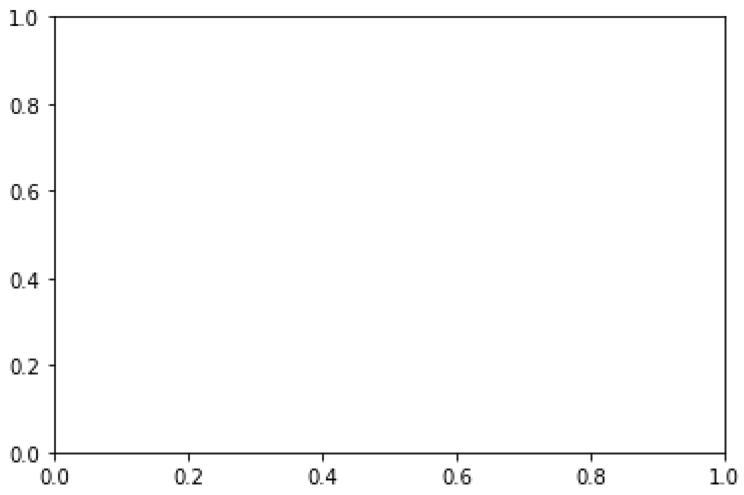


```
sbn.scatterplot(x=dataset['is_late_to_pay'],y=dataset['payment_full'], title='Relationship be
```

140

```
"""
The one to one relationship between late payment and partial. This is that all the individual
"""
```

```
---------------------------------------------------------------------------

AttributeError                            Traceback (most recent call last)

/var/folders/2r/mmf04xt95z9g0hpl52c0vyj80000gn/T/ipykernel_4025/2939396357.py in <module>
----> 1 sbn.scatterplot(x=dataset['is_late_to_pay'],y=dataset['payment_full'], title='Relatio
      2 
      3 """
      4 The one to one relationship between late payment and partial. This is that all the in
      5 """

~/Documents/watu_credit_data_interview/venv/lib/python3.8/site-packages/seaborn/_decorators.p
     44             )
     45         kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
---> 46         return f(**kwargs)
     47     return inner_f
     48 

~/Documents/watu_credit_data_interview/venv/lib/python3.8/site-packages/seaborn/relational.py
    825     p._attach(ax)
    826 
--> 827     p.plot(ax, kwargs)
    828 
    829     return ax

~/Documents/watu_credit_data_interview/venv/lib/python3.8/site-packages/seaborn/relational.py
    606         )
    607         scout_x = scout_y = np.full(scout_size, np.nan)
--> 608         scout = ax.scatter(scout_x, scout_y, **kws)
    609         s = kws.pop("s", scout.get_sizes())
    610         c = kws.pop("c", scout.get_facecolors())

~/Documents/watu_credit_data_interview/venv/lib/python3.8/site-packages/matplotlib/__init__.p
   1359     def inner(ax, *args, data=None, **kwargs):
   1360         if data is None:
-> 1361             return func(ax, *map(sanitize_sequence, args), **kwargs)
   1362 
   1363         bound = new_sig.bind(ax, *args, **kwargs)

~/Documents/watu_credit_data_interview/venv/lib/python3.8/site-packages/matplotlib/axes/_axes
   4595                 )
   4596         collection.set_transform(mtransforms.IdentityTransform())
-> 4597         collection.update(kwargs)
   4598 
   4599         if colors is None:

~/Documents/watu_credit_data_interview/venv/lib/python3.8/site-packages/matplotlib/artist.py
   1060                     func = getattr(self, f"set_{k}", None)
   1061                     if not callable(func):
-> 1062                         raise AttributeError(f"{type(self).__name__!r} object "
   1063                                              f"has no property {k!r}")
   1064                     ret.append(func(v))

AttributeError: 'PathCollection' object has no property 'title'
```


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0