Supermarket sales data analysis using Python

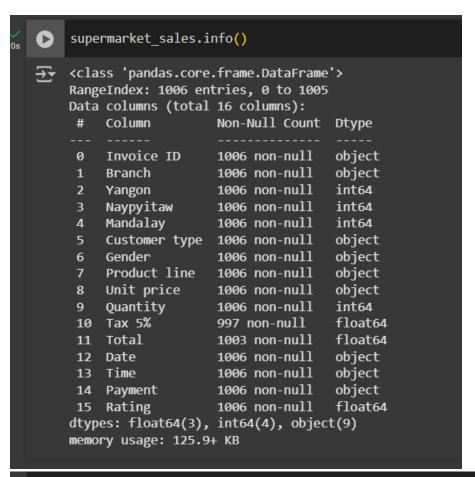
1-Gather Data

In this step we gather the Excel file "Python Project Data – Supermarket sale (2)



2-Assess

In this step we do an assessment to data to explore the both Quality and Tidness issues with the data to reach an accurate analysis



os	[3]	<pre>supermarket_sales.describe()</pre>									
	₹		Yangon	Naypyitaw	Mandalay	Quantity	Tax 5%	Total	Rating		
		count	1006.000000	1006.000000	1006.000000	1006.000000	997.000000	1003.000000	1006.000000		
		mean	0.338966	0.329026	0.332008	5.469185	15.479682	322.734689	7.056163		
		std	0.473594	0.470093	0.471168	3.014153	11.728320	245.865964	3.318751		
		min	0.000000	0.000000	0.000000	-8.000000	0.508500	10.678500	4.000000		
		25%	0.000000	0.000000	0.000000	3.000000	5.986500	123.789750	5.500000		
		50%	0.000000	0.000000	0.000000	5.000000	12.227500	254.016000	7.000000		
		75%	1.000000	1.000000	1.000000	8.000000	22.720500	471.009000	8.500000		
		max	1.000000	1.000000	1.000000	10.000000	49.650000	1042.650000	97.000000		

By doing an overall review of the data we can see some

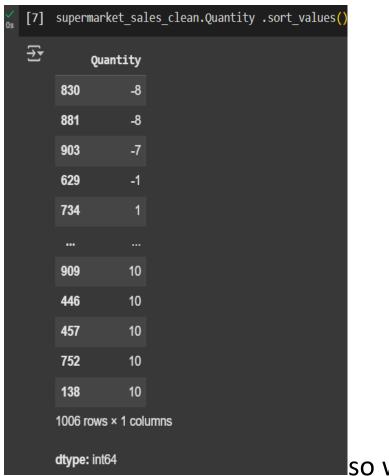
Quality issues :

- Missing values in both Total and Tax 5% (but we can calculate)
- Outliers in Rating values "there are one values more than 10(97)" (but we can fix)
- Inconsistent data in Time "one value is 8-30 pm " (but we can fix it)
- Inconsistent data in Unit price " 5 values have the price with USD" (but we can fix)
- Invalid data with Quantity "3 values have a negative number" (but we can fix)
- Invalid data with Date
- Tidiness issues
- The three columns "Yangon"," Naypyitaw" and "Mandalay" must be in one column called (city)
 - 3- Cleaning

In this step we fix the quality issues existing in the data to start making the analysis So we will take a copy of the original data

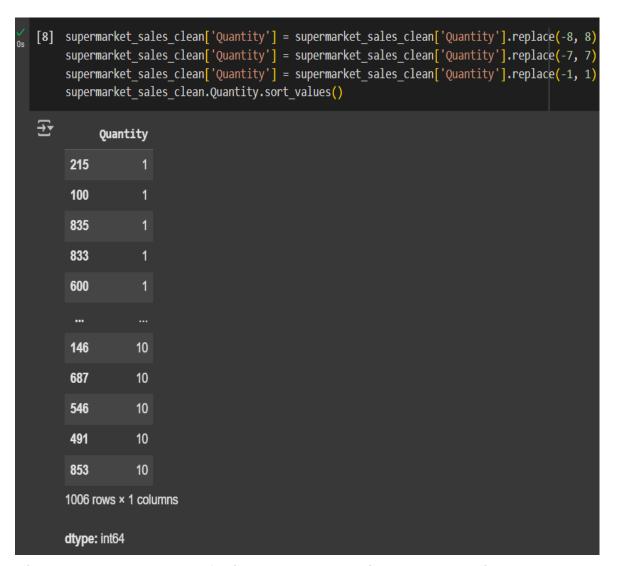


We will firstly fix the issues of negative Quantity values:



lso we will

replace the negative numbers with a positive ones



then we ensured this step with seeing the sorted values again

The next step we did that we removed the USD within the Unit price column to make it ready for analysis

```
[10] supermarket_sales_clean.loc[supermarket_sales_clean['Invoice ID']=='865-41-9075', 'Unit price']=11.53
#print(supermarket_sales_clean.iloc[903])
supermarket_sales_clean.loc[supermarket_sales_clean['Invoice ID']=='115-38-7388', 'Unit price']=10.18
#print(supermarket_sales_clean.iloc[881])
supermarket_sales_clean.loc[supermarket_sales_clean['Invoice ID']=='237-44-6163', 'Unit price']=10.56
#print(supermarket_sales_clean.iloc[830])
supermarket_sales_clean.loc[supermarket_sales_clean['Invoice ID']=='308-39-1707', 'Unit price']=12.09
#print(supermarket_sales_clean.iloc[629])
supermarket_sales_clean.loc[supermarket_sales_clean['Invoice ID']==' 871-39-9221', 'Unit price']=12.45
#print(supermarket_sales_clean.iloc[629])
```

We then showed the null cells in both Total and Tax 5%

(_s [5]	supe	rmarket_sale	s_clean[supermar	ket_sales_o	clean['Tota	al'].isnull()]										
		Invoice ID	Branch	Yangon	Naypyitaw	Mandalay	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time Pay	ment Rating	
	0	750-67-8428	A		0	0	Normal	Male	Health and beauty	74.69	7	26.1415	NaN	1/5/2019	13:08 Ev	rallet 9.1	
	14	829-34-3910	A		0	0	Normal	Male	Health and beauty	71.38	10	35.6900	NaN	3/29/2019	19:21	Cash 5.7	
	37	272-65-1806	A		0	0	Normal	Male	Electronic accessories	60.88	9	27.3960	NaN	1/15/2019	17:17 Ev	rallet 4.7	
US L 3	supe	rmarket_sale	s_clean[supermar	ket_sales_o	:lean['Tax	5%'].isnull()]										
<u>-</u> }		Invoice ID	Branch	Yangon	Naypyitaw	Mandalay	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Tota	ıl Dat	e Tin	e Payment	Rating
	3	123-19-1176	A		0	0	Normal	Male	Health and beauty	58.22	! 8	NaN	489.048	0 1/27/201	9 8-30P	M Ewalle	8.4
	8	665-32-9167	A	. 1	0	0	Normal	Male	Health and beauty	36.26	2	NaN	76.146	0 1/10/201	9 17:1	5 Credit card	7.2
	86	362-58-8315	C	0	1	0	Normal	Male	Fashion accessories	76.52	! 5	NaN	401.730	0 3/25/201	9 10:2	3 Cash	9.9
	92	873-51-0671	A	. 1	0	0	Member	Female	Sports and travel	21.98	7	NaN	161.553	0 1/10/201	9 16:4	2 Ewalle	5.1
	97	871-39-9221	C	0	1	0	Normal	Female	Electronic accessories	12.45 USD	6	NaN	78.435	0 2/9/201	9 13:	1 Cash	4.1
	629	308-39-1707	A	. 1	0	0	Normal	Female	Fashion accessories	12.09 USD	-1	NaN	12.694	5 1/26/201	9 18:1	9 Credit card	8.2
	830	237-44-6163	A	1	0	0	Normal	Male	Electronic accessories	10.56 USD	3- (NaN	88.704	0 1/24/201	9 17:4	3 Cash	7.6
	881	115-38-7388	C	0	1	0	Member	Female	Fashion accessories	10.18 USD	3- (NaN	85.512	0 3/30/201	9 12:5	1 Credit card	9.5
	903	865-41-9075	A		0	0	Normal	Male	Food and beverages	11.53 USD	-7	NaN	84.745	5 1/28/201	9 17:3	5 Cash	8.1

To avoid getting nulls in the two columns we entered the calculated values of Total

```
import numpy as np

supermarket_sales_clean.loc[supermarket_sales_clean['Invoice ID']=='750-67-8428', 'Total']=548.9715
supermarket_sales_clean.loc[supermarket_sales_clean['Invoice ID']=='829-34-3910', 'Total']=749.49
supermarket_sales_clean.loc[supermarket_sales_clean['Invoice ID']=='272-65-1806', 'Total']=575.316
supermarket_sales_clean.loc[supermarket_sales_clean['Invoice ID']=='272-65-1806', 'Total']=575.316
supermarket_sales_clean.info()
```

after this we defined Tax 5% as Total – "Unit price * Quantity "

```
[19] supermarket_sales_clean['Tax 5%'] = supermarket_sales_clean['Total'] - supermarket_sales_clean['Unit price'] * supermarket_sales_clean['Quantity'] supermarket_sales_clean.info()
```

Then we ensured that the are no nulls in the

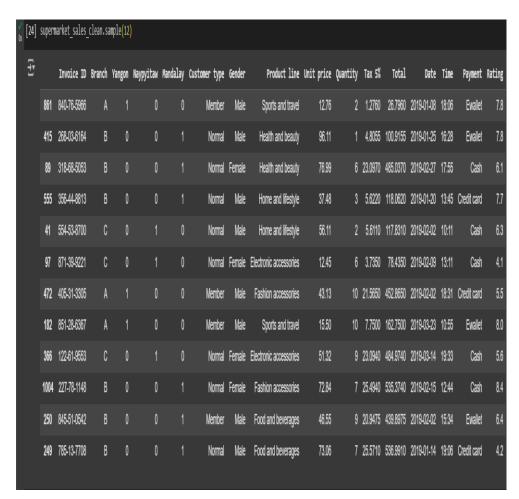
```
<class 'pandas.core.frame.DataFrame'>
 RangeIndex: 1006 entries, 0 to 1005
 Data columns (total 16 columns):
                   Non-Null Count Dtype
     Column
     Invoice ID
                   1006 non-null
                                   object
  0
  1
     Branch
                   1006 non-null
                                   object
  2
     Yangon
                   1006 non-null
                                   int64
     Naypyitaw
                   1006 non-null
                                   int64
 4
     Mandalay
                   1006 non-null int64
  5
     Customer type 1006 non-null
                                   object
  6
     Gender
                   1006 non-null
                                   object
     Product line 1006 non-null
  7
                                   object
  8
     Unit price
                   1006 non-null
                                   float64
     Quantity
                   1006 non-null
                                   int64
 9
  10 Tax 5%
                   1006 non-null
                                   float64
                   1006 non-null float64
  11 Total
  12 Date
                   1006 non-null
                                   obiect
  13 Time
                   1006 non-null
                                   object
  14 Payment
                   1006 non-null
                                   object
                  1006 non-null
                                   float64
  15 Rating
 dtypes: float64(4), int64(4), object(8)
 memory usage: 125.9+ KB
```

data then we

fixed the issue of Date

```
supermarket_sales_clean['Date'] = pd.to_datetime(supermarket_sales_clean['Date'])
    supermarket sales clean.info()
    supermarket_sales_clean.head()
→ <class 'pandas.core.frame.DataFrame'>
    RangeIndex: 1006 entries, 0 to 1005
    Data columns (total 16 columns):
        Column
                      Non-Null Count Dtype
         Invoice ID
                      1006 non-null object
        Branch
                      1006 non-null
                                     object
     2 Yangon
                      1006 non-null int64
     3 Naypyitaw
                      1006 non-null int64
     4 Mandalay
                      1006 non-null int64
     5 Customer type 1006 non-null object
     6 Gender
                      1006 non-null object
     7 Product line 1006 non-null object
     8 Unit price
                      1006 non-null float64
     9 Quantity
                      1006 non-null int64
     10 Tax 5%
                      1006 non-null float64
     11 Total
                    1006 non-null float64
     12 Date
                      1006 non-null datetime64[ns]
     13 Time
                    1006 non-null object
                 1006 non-null object
1006 non-null float64
     14 Payment
    dtypes: datetime64[ns](1), float64(4), int64(4), object(7)
    memory usage: 125.9+ KB
                                                                                we
```

took a sample with size 12 to see the change



then we moved to the next issue with Time that one value don't match the column "8-30 pm "

So we changed the time type

```
import pandas as pd

# to change the type of time to match all the data frame
supermarket_sales_clean['Time'] = pd.to_datetime(supermarket_sales['Time'], errors='coerce').dt.strftime('%H:%M')

print(supermarket_sales_clean[['Time']].head())

Time
0 13:08
1 10:29
2 13:23
3 NaN
4 10:37
```

so we got null in the cell mentiondbut we can fix this null to 20:30

```
[28] supermarket_sales_clean['Time'] = supermarket_sales_clean['Time'].fillna('20:30')
     supermarket sales clean[supermarket sales clean['Time'].isnull()]
     supermarket_sales_clean.info()
→ <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 1006 entries, 0 to 1005
     Data columns (total 16 columns):
                         Non-Null Count Dtype
      # Column
          Invoice ID
                         1006 non-null object
      0
          Branch
                         1006 non-null object
          Yangon
                         1006 non-null int64
          Naypyitaw
                         1006 non-null int64
      4 Mandalay
                         1006 non-null int64
      5 Customer type 1006 non-null object
      6 Gender
                         1006 non-null object
          Product line 1006 non-null object
         Unit price
                         1006 non-null float64
          Quantity
                         1006 non-null int64
                      1006 non-null float64
1006 non-null float64
1006 non-null datetime64[ns]
1006 non-null object
1006 non-null object
      10 Tax 5%
      11 Total
      12 Date
      13 Time
      14 Payment
      15 Rating
                         1006 non-null float64
     dtypes: datetime64[ns](1), float64(4), int64(4), object(7)
     memory usage: 125.9+ KB
```

now we don't have nulls

Then we moved to the last issue with a more than 10 value

```
[31] print(supermarket_sales_clean.iloc[157])
   → Invoice ID
                                307-85-2293
       Branch
                                          В
       Yangon
                                          0
       Naypyitaw
                                          0
       Mandalay
                                          1
       Customer type
                                     Normal
       Gender
                                       Male
                       Home and lifestyle
       Product line
       Unit price
                                      50.28
       Quantity
       Tax 5%
                                      12.57
       Total
                                     263.97
       Date
                        2019-03-07 00:00:00
       Time
                                      13:58
       Payment
                                    Ewallet
       Rating
                                       97.0
       Name: 157, dtype: object
```

Then we fixed this error

```
[33] supermarket_sales_clean['Rating'] = supermarket_sales_clean['Rating'].replace(97, 9.7)
     print(supermarket_sales_clean.iloc[157])
→ Invoice ID
                             307-85-2293
     Branch
                                       В
     Yangon
                                       0
     Naypyitaw
                                       0
     Mandalay
     Customer type
                                  Normal
     Gender
                                    Male
     Product line
                      Home and lifestyle
     Unit price
                                   50.28
     Quantity
     Tax 5%
                                   12.57
     Total
                                  263.97
     Date
                     2019-03-07 00:00:00
     Time
                                   13:58
     Payment
                                 Ewallet
     Rating
                                     9.7
    Name: 157, dtype: object
```

```
supermarket_sales_clean.info()
<<class 'pandas.core.frame.DataFrame'>
    RangeIndex: 1006 entries, 0 to 1005
    Data columns (total 16 columns):
                       Non-Null Count Dtype
         Column
         Invoice ID
                                       object
     0
                       1006 non-null
     1
         Branch
                       1006 non-null
                                       object
         Yangon
                       1006 non-null
                                       int64
         Naypyitaw
                       1006 non-null
                                       int64
                       1006 non-null
                                       int64
     4
         Mandalay
         Customer type 1006 non-null
                                       object
     6
         Gender
                       1006 non-null
                                       object
         Product line
                       1006 non-null
                                       object
                                       float64
     8
         Unit price
                       1006 non-null
     9
         Quantity
                       1006 non-null
                                       int64
     10 Tax 5%
                       1006 non-null
                                       float64
     11 Total
                       1006 non-null
                                       float64
                                       datetime64[ns]
     12
        Date
                       1006 non-null
        Time
     13
                       1006 non-null
                                       object
     14 Payment
                       1006 non-null
                                       object
     15 Rating
                       1006 non-null
                                       float64
    dtypes: datetime64[ns](1), float64(4), int64(4), object(7)
    memory usage: 125.9+ KB
```

√ 0s	[38]	<pre>supermarket_sales_clean.describe()</pre>											
	- }•	Yangon		Naypyitaw	Mandalay	Unit price Quantity		Tax 5% Total		Date	Rating		
		count	1006.000000	1006.000000	1006.000000	1006.000000	1006.000000	1006.000000	1006.000000	1006	1006.000000		
		mean	0.338966	0.329026	0.332008	55.699493	5.516899	14.917266	323.140944	2019-02-14 00:22:54.155069440	6.969384		
		min	0.000000	0.000000	0.000000	10.080000	1.000000	-469.485000	10.678500	2019-01-01 00:00:00	4.000000		
		25%	0.000000	0.000000	0.000000	32.975000	3.000000	5.864625	123.157125	2019-01-24 00:00:00	5.500000		
		50%	0.000000	0.000000	0.000000	55.420000	5.000000	12.123000	254.583000	2019-02-13 00:00:00	7.000000		
		75%	1.000000	1.000000	1.000000	77.945000	8.000000	22.475750	471.990750	2019-03-08 00:00:00	8.500000		
		max	1.000000	1.000000	1.000000	99.960000	10.000000	49.650000	1042.650000	2019-03-30 00:00:00	10.000000		
		std	0.473594	0.470093	0.471168	26.505795	2.925818	19.258809	246.091420	NaN	1.721592		

We created the column of city to fix Tidiness issue



Now the data is ready for the analysis