Supermarket sales data analysis using Python

1-Gather Data

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

```
[1] import pandas as pd
    supermarket_sales = pd.read_csv("/content/Python Project Data - Supermarket Sales (2).csv")
    supermarket_sales.head()
```

	Invoice ID	Branch	Yangon	Naypyitaw	Mandalay	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	Rating
0	750-67-8428	A	1	0	0	Normal	Male	Health and beauty	74.69	7	26.1415	NaN	1/5/2019	13:08	Ewallet	9.1
1	226-31-3081	C	0	1	0	Normal	Male	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29	Cash	9.6
2	631-41-3108	A	1	0	0	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	13:23	Credit card	7.4
3	123-19-1176	A	1	0	0	Normal	Male	Health and beauty	58.22	8	NaN	489.0480	1/27/2019	8 - 30 PM	Ewallet	8.4
4	373-73-7910	A	1	0	0	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37	Ewallet	5.3

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

```
supermarket_sales.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1006 entries, 0 to 1005
Data columns (total 16 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Invoice ID     1006 non-null   object
 1   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   object
 9   Quantity       1006 non-null   int64
 10  Tax 5%         997 non-null    float64
 11  Total          1003 non-null   float64
 12  Date           1006 non-null   object
 13  Time           1006 non-null   object
 14  Payment        1006 non-null   object
 15  Rating         1006 non-null   float64
dtypes: float64(3), int64(4), object(9)
memory usage: 125.9+ KB
```

```
[3] supermarket_sales.describe()
```

	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

```
[4] supermarket_sales_clean= supermarket_sales.copy()
    supermarket_sales_clean.head()
```

	Invoice ID	Branch	Yangon	Naypyitaw	Mandalay	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	Rating
0	750-67-8428	A	1	0	0	Normal	Male	Health and beauty	74.69	7	26.1415	NaN	1/5/2019	13:08	Ewallet	9.1
1	226-31-3081	C	0	1	0	Normal	Male	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29	Cash	9.6
2	631-41-3108	A	1	0	0	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	13:23	Credit card	7.4
3	123-19-1176	A	1	0	0	Normal	Male	Health and beauty	58.22	8	NaN	489.0480	1/27/2019	8 - 30 PM	Ewallet	8.4
4	373-73-7910	A	1	0	0	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37	Ewallet	5.3

We will firstly fix the issues of negative Quantity values:

```
[7] supermarket_sales_clean.Quantity .sort_values()
```

	Quantity
830	-8
881	-8
903	-7
629	-1
734	1
...	...
909	10
446	10
457	10
752	10
138	10

1006 rows × 1 columns

dtype: int64

so we will replace the negative numbers with a positive ones

```
[8] supermarket_sales_clean['Quantity'] = supermarket_sales_clean['Quantity'].replace(-8, 8)
    supermarket_sales_clean['Quantity'] = supermarket_sales_clean['Quantity'].replace(-7, 7)
    supermarket_sales_clean['Quantity'] = supermarket_sales_clean['Quantity'].replace(-1, 1)
    supermarket_sales_clean.Quantity.sort_values()
```

	Quantity
215	1
100	1
835	1
833	1
600	1
...	...
146	10
687	10
546	10
491	10
853	10

1006 rows × 1 columns

dtype: int64

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%

```
[5] supermarket_sales_clean[supermarket_sales_clean['Total'].isnull()]
```

	Invoice ID	Branch	Yangon	Naypyitaw	Mandalay	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	Rating
0	750-67-8428	A	1	0	0	Normal	Male	Health and beauty	74.69	7	26.1415	NaN	1/5/2019	13:08	Ewallet	9.1
14	829-34-3910	A	1	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	1	0	0	Normal	Male	Electronic accessories	60.88	9	27.3960	NaN	1/15/2019	17:17	Ewallet	4.7

```
[6] supermarket_sales_clean[supermarket_sales_clean['Tax 5%'].isnull()]
```

	Invoice ID	Branch	Yangon	Naypyitaw	Mandalay	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	Rating
3	123-19-1176	A	1	0	0	Normal	Male	Health and beauty	58.22	8	NaN	489.0480	1/27/2019	8 - 30 PM	Ewallet	8.4
8	665-32-9167	A	1	0	0	Normal	Male	Health and beauty	36.26	2	NaN	76.1460	1/10/2019	17:15	Credit card	7.2
86	362-58-8315	C	0	1	0	Normal	Male	Fashion accessories	76.52	5	NaN	401.7300	3/25/2019	10:23	Cash	9.9
92	873-51-0671	A	1	0	0	Member	Female	Sports and travel	21.98	7	NaN	161.5530	1/10/2019	16:42	Ewallet	5.1
97	871-39-9221	C	0	1	0	Normal	Female	Electronic accessories	12.45 USD	6	NaN	78.4350	2/9/2019	13:11	Cash	4.1
629	308-39-1707	A	1	0	0	Normal	Female	Fashion accessories	12.09 USD	-1	NaN	12.6945	1/26/2019	18:19	Credit card	8.2
830	237-44-6163	A	1	0	0	Normal	Male	Electronic accessories	10.56 USD	-8	NaN	88.7040	1/24/2019	17:43	Cash	7.6
881	115-38-7388	C	0	1	0	Member	Female	Fashion accessories	10.18 USD	-8	NaN	85.5120	3/30/2019	12:51	Credit card	9.5
903	865-41-9075	A	1	0	0	Normal	Male	Food and beverages	11.53 USD	-7	NaN	84.7455	1/28/2019	17:35	Cash	8.1

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

```
[18] #to exchange the NaN cell in the first cell we will calculate it
     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 data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1006 entries, 0 to 1005
Data columns (total 16 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Invoice ID     1006 non-null   object
 1   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   object
 13  Time           1006 non-null   object
 14  Payment        1006 non-null   object
 15  Rating         1006 non-null   float64
dtypes: float64(4), int64(4), object(8)
memory usage: 125.9+ KB
```

then we fixed the issue of Date

```
#to fix the issue with the 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
---  ------         --------------  -----
 0   Invoice ID     1006 non-null   object
 1   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
 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
```

we took a sample with size 12 to see the change

```
[24] supermarket_sales_clean.sample(12)
```

	Invoice ID	Branch	Yangon	Naypyitaw	Mandalay	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	Rating
861	840-76-5966	A	1	0	0	Member	Male	Sports and travel	12.76	2	1.2760	26.7960	2019-01-08	18:06	Ewallet	7.8
415	268-03-6164	B	0	0	1	Normal	Male	Health and beauty	96.11	1	4.8055	100.9155	2019-01-25	16:28	Ewallet	7.8
89	318-68-5053	B	0	0	1	Normal	Female	Health and beauty	76.99	6	23.0970	485.0370	2019-02-27	17:55	Cash	6.1
555	356-44-8813	B	0	0	1	Normal	Male	Home and lifestyle	37.48	3	5.6220	118.0620	2019-01-20	13:45	Credit card	7.7
41	554-53-8700	C	0	1	0	Normal	Male	Home and lifestyle	56.11	2	5.6110	117.8310	2019-02-02	10:11	Cash	6.3
97	871-39-9221	C	0	1	0	Normal	Female	Electronic accessories	12.45	6	3.7350	78.4350	2019-02-09	13:11	Cash	4.1
472	405-31-3305	A	1	0	0	Member	Male	Fashion accessories	43.13	10	21.5650	452.8650	2019-02-02	18:31	Credit card	5.5
182	851-28-6367	A	1	0	0	Member	Male	Sports and travel	15.50	10	7.7500	162.7500	2019-03-23	10:55	Ewallet	8.0
366	122-61-9563	C	0	1	0	Normal	Female	Electronic accessories	51.32	9	23.0940	484.9740	2019-03-14	19:33	Cash	5.6
1004	227-78-1148	B	0	0	1	Normal	Female	Fashion accessories	72.84	7	25.4940	535.3740	2019-02-15	12:44	Cash	8.4
250	845-51-0542	B	0	0	1	Member	Male	Food and beverages	46.55	9	20.9475	439.8975	2019-02-02	15:34	Ewallet	6.4
249	785-13-7708	B	0	0	1	Normal	Male	Food and beverages	73.06	7	25.5710	536.9910	2019-01-14	19:06	Credit card	4.2

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):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Invoice ID     1006 non-null   object
 1   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
 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
```

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                              B
Yangon                              0
Naypyitaw                           0
Mandalay                            1
Customer type                  Normal
Gender                           Male
Product line      Home and lifestyle
Unit price                     50.28
Quantity                           5
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                            B
Yangon                            0
Naypyitaw                         0
Mandalay                          1
Customer type                Normal
Gender                         Male
Product line      Home and lifestyle
Unit price                    50.28
Quantity                          5
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):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Invoice ID     1006 non-null   object
 1   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
 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
```

```
[38] supermarket_sales_clean.describe()
```

	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

```
[173] market_sales_clean=pd.melt(supermarket_sales_clean,id_vars=['Invoice ID','Branch','Customer type','Gender','Product line','Unit price','Quantity','Tax 5%','Total','Date','Time',
                                                                  'Payment','Rating'],var_name='city',value_name='city_value')
```

```
[174] supermarket_sales_clean.head()
```

	Invoice ID	Branch	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	Rating	city	city_value
0	750-67-8428	A	Normal	Male	Health and beauty	74.69	7	26.1415	548.9715	2019-01-05	13:08	Ewallet	9.1	Yangon	1
1	226-31-3081	C	Normal	Male	Electronic accessories	15.28	5	3.8200	80.2200	2019-03-08	10:29	Cash	9.6	Yangon	0
2	631-41-3108	A	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	2019-03-03	13:23	Credit card	7.4	Yangon	1
3	123-19-1176	A	Normal	Male	Health and beauty	58.22	8	23.2880	489.0480	2019-01-27	20:30	Ewallet	8.4	Yangon	1
4	373-73-7910	A	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2019-02-08	10:37	Ewallet	5.3	Yangon	1

Now the data is ready for the analysis