In [1]:

```
import pandas as pd
```

In [2]:

```
sales_2017=pd.read_csv('Sales Transactions-2017.csv')
sales_2017
```

Out[2]:

	Date	Voucher	Party	Product	Qty	Rate	Gr
0	1/4/2017	Sal:1	SOLANKI PLASTICS	DONA-VAI-9100	2	1,690.00	3,38(
1	1/4/2017	Sal:1	SOLANKI PLASTICS	LITE FOAM(1200)	6	1,620.00	9,72(
2	1/4/2017	Sal:2	SARNESWARA TRADERS	VISHNU CHOTA WINE	500	23	11,50(
3	1/4/2017	Sal:2	SARNESWARA TRADERS	LITE FOAM(1200)	6	1,620.00	9,72(
4	1/4/2017	Sal:2	SARNESWARA TRADERS	DONA-VAI-9100	5	1,690.00	8,45(
...	...	...	...	...	...	...	
47285	31/03/2018	Sal:10042	Vkp	10*10 SHEET	25	137	3,42
47286	NaN	NaN	NaN	NaN	NaN	NaN	N
47287	NaN	NaN	NaN	NaN	NaN	NaN	N
47288	NaN	Total	NaN	NaN	607,734.60	669,300.49	9,953,81(
47289	NaN	Total	NaN	NaN	7,593,062.00	8,309,116.00	115,778,72

47290 rows × 9 columns

In [3]:

```
sales_2018=pd.read_csv('Sales Transactions-2018.csv')
sales_2018
```

Out[3]:

	Date	Voucher	Party	Product	Qty	Rate	Gross	
0	1/4/2018	Sal:146	TP13	SILVER POUCH 9*12	50	85	4,250.00	
1	1/4/2018	Sal:146	TP13	RUBBER	5	290	1,450.00	
2	1/4/2018	Sal:146	TP13	DURGA 10*12 Blue	1,600.00	5.5	8,800.00	
3	1/4/2018	Sal:146	TP13	DURGA 13*16 BLUE	400	11	4,400.00	
4	1/4/2018	Sal:146	TP13	10*12 SARAS-NAT	600	8.1	4,860.00	
...	...	...	...	...	...	...	...	
44735	31/03/2019	Sal:9610	HAMPI FOODS	SPOON SOOFY	200	40	8,000.00	
44736	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
44737	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
44738	NaN	Total	NaN	NaN	666,056.00	1,067,808.80	10,796,991.30	29,9
44739	NaN	Total	NaN	NaN	7,097,803.00	10,024,197.00	117,897,671.80	720,2

44740 rows × 9 columns

In [4]:

```
sales_2019=pd.read_csv('Sales Transactions-2019.csv')
sales_2019
```

Out[4]:

	Date	Voucher	Party	Product	Qty	Rate	Gross	
0	1/4/2019	Sal:687	BALAJI PLASTICS	DONA-VAI-9100	1	1,730.00	1,730.00	
1	1/4/2019	Sal:687	BALAJI PLASTICS	SMART BOUL(48)	1	1,730.00	1,730.00	
2	1/4/2019	Sal:688	BALAJI PLASTICS	Vishnu Ice	110	18.5	2,035.00	
3			28/3		0	0		
4	1/4/2019	Sal:689	BALAJI PLASTICS	100LEAF-SP	3	585	1,755.00	
...	...	...	...	...	...	...	...	
19171	10/10/2019	Sal:4935	K.SRIHARI	13*16 WHITE RK	400	16	6,400.00	
19172	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
19173	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
19174	NaN	Total	NaN	NaN	99,284.90	175,381.65	2,203,649.50	2(
19175	NaN	Total	NaN	NaN	2,710,193.00	5,519,888.40	53,360,791.40	672

19176 rows × 9 columns

In [5]:

```
sales_2017.shape,sales_2018.shape,sales_2019.shape
```

Out[5]:

```
((47290, 9), (44740, 9), (19176, 9))
```

In [6]:

```
sales_complete_data=pd.concat([sales_2017,sales_2018,sales_2019])
sales_complete_data
```

Out[6]:

	Date	Voucher	Party	Product	Qty	Rate	Gros
0	1/4/2017	Sal:1	SOLANKI PLASTICS	DONA-VAI-9100	2	1,690.00	3,380.0
1	1/4/2017	Sal:1	SOLANKI PLASTICS	LITE FOAM(1200)	6	1,620.00	9,720.0
2	1/4/2017	Sal:2	SARNESWARA TRADERS	VISHNU CHOTA WINE	500	23	11,500.0
3	1/4/2017	Sal:2	SARNESWARA TRADERS	LITE FOAM(1200)	6	1,620.00	9,720.0
4	1/4/2017	Sal:2	SARNESWARA TRADERS	DONA-VAI-9100	5	1,690.00	8,450.0
...	...	...	...	...	...	...	
19171	10/10/2019	Sal:4935	K.SRIHARI	13*16 WHITE RK	400	16	6,400.0
19172	NaN	NaN	NaN	NaN	NaN	NaN	Na
19173	NaN	NaN	NaN	NaN	NaN	NaN	Na
19174	NaN	Total	NaN	NaN	99,284.90	175,381.65	2,203,649.5
19175	NaN	Total	NaN	NaN	2,710,193.00	5,519,888.40	53,360,791.4

111206 rows × 9 columns

Initial analysis:

In [7]:

```
sales_complete_data.shape
```

Out[7]:

```
(111206, 9)
```

In [8]:

```
sales_complete_data.isnull().sum()
```

Out[8]:

```
Date              12591
Voucher           12557
Party                40
Product           12591
Qty               12557
Rate              12558
Gross             12558
Disc             105609
Voucher Amount    83646
dtype: int64
```

In [9]:

```
sales_complete_data.isnull().sum().sum()                                    #total null values
```

Out[9]:

```
264707
```

In [10]:

```
sales_complete_data.info
```

Out[10]:

```
<bound method DataFrame.info of             Date   Voucher               Pa
rty            Product  \
0        1/4/2017     Sal:1    SOLANKI PLASTICS      DONA-VAI-9100
1        1/4/2017     Sal:1    SOLANKI PLASTICS    LITE FOAM(1200)
2        1/4/2017     Sal:2  SARNESWARA TRADERS  VISHNU CHOTA WINE
3        1/4/2017     Sal:2  SARNESWARA TRADERS    LITE FOAM(1200)
4        1/4/2017     Sal:2  SARNESWARA TRADERS      DONA-VAI-9100
...           ...       ...                 ...                 ...
19171  10/10/2019  Sal:4935           K.SRIHARI     13*16 WHITE RK
19172         NaN       NaN                 NaN                 NaN
19173         NaN       NaN                 NaN                 NaN
19174         NaN   Total                   NaN                 NaN
19175         NaN   Total                   NaN                 NaN

                Qty          Rate          Gross        Disc Voucher Amount
0                 2      1,690.00       3,380.00         NaN      13,100.00
1                 6      1,620.00       9,720.00         NaN            NaN
2               500            23      11,500.00         NaN      30,990.00
3                 6      1,620.00       9,720.00         NaN            NaN
4                 5      1,690.00       8,450.00         NaN            NaN
...             ...           ...           ...         ...            ...
19171           400            16       6,400.00         NaN            NaN
19172           NaN           NaN            NaN         NaN            NaN
19173           NaN           NaN            NaN         NaN            NaN
19174     99,284.90    175,381.65   2,203,649.50   20,680.00   2,189,014.50
19175  2,710,193.00  5,519,888.40  53,360,791.40  672,984.00  52,830,224.40

[111206 rows x 9 columns]>
```

In [11]:

```
sales_complete_data.describe(include='all')
```

Out[11]:

	Date	Voucher	Party	Product	Qty	Rate	Gross	Disc	Voucher Amount
count	98615	98649	111166	98615	98649	98648	98648	5597	27560
unique	836	10044	1994	867	546	1109	2548	371	7149
top			TP13		100				
freq	3053	3053	13056	3053	12528	3051	3053	3053	3053

In [12]:

```
sales_complete_data.dtypes
```

Out[12]:

```
Date              object
Voucher           object
Party             object
Product           object
Qty               object
Rate              object
Gross             object
Disc              object
Voucher Amount    object
dtype: object
```

In [13]:

```
sales_complete_data.any()                                    #it returns true if a
```

Out[13]:

```
Date              True
Voucher           True
Party             True
Product           True
Qty               True
Rate              True
Gross             True
Disc              True
Voucher Amount    True
dtype: bool
```

In [14]:

```
sales_complete_data.head()
```

Out[14]:

	Date	Voucher	Party	Product	Qty	Rate	Gross	Disc	Voucher Amount
0	1/4/2017	Sal:1	SOLANKI PLASTICS	DONA-VAI-9100	2	1,690.00	3,380.00	NaN	13,100.00
1	1/4/2017	Sal:1	SOLANKI PLASTICS	LITE FOAM(1200)	6	1,620.00	9,720.00	NaN	NaN
2	1/4/2017	Sal:2	SARNESWARA TRADERS	VISHNU CHOTA WINE	500	23	11,500.00	NaN	30,990.00
3	1/4/2017	Sal:2	SARNESWARA TRADERS	LITE FOAM(1200)	6	1,620.00	9,720.00	NaN	NaN
4	1/4/2017	Sal:2	SARNESWARA TRADERS	DONA-VAI-9100	5	1,690.00	8,450.00	NaN	NaN

In [15]:

```
import pandas as pd
```

In [16]:

```
import matplotlib.pyplot as plt
```

In [17]:

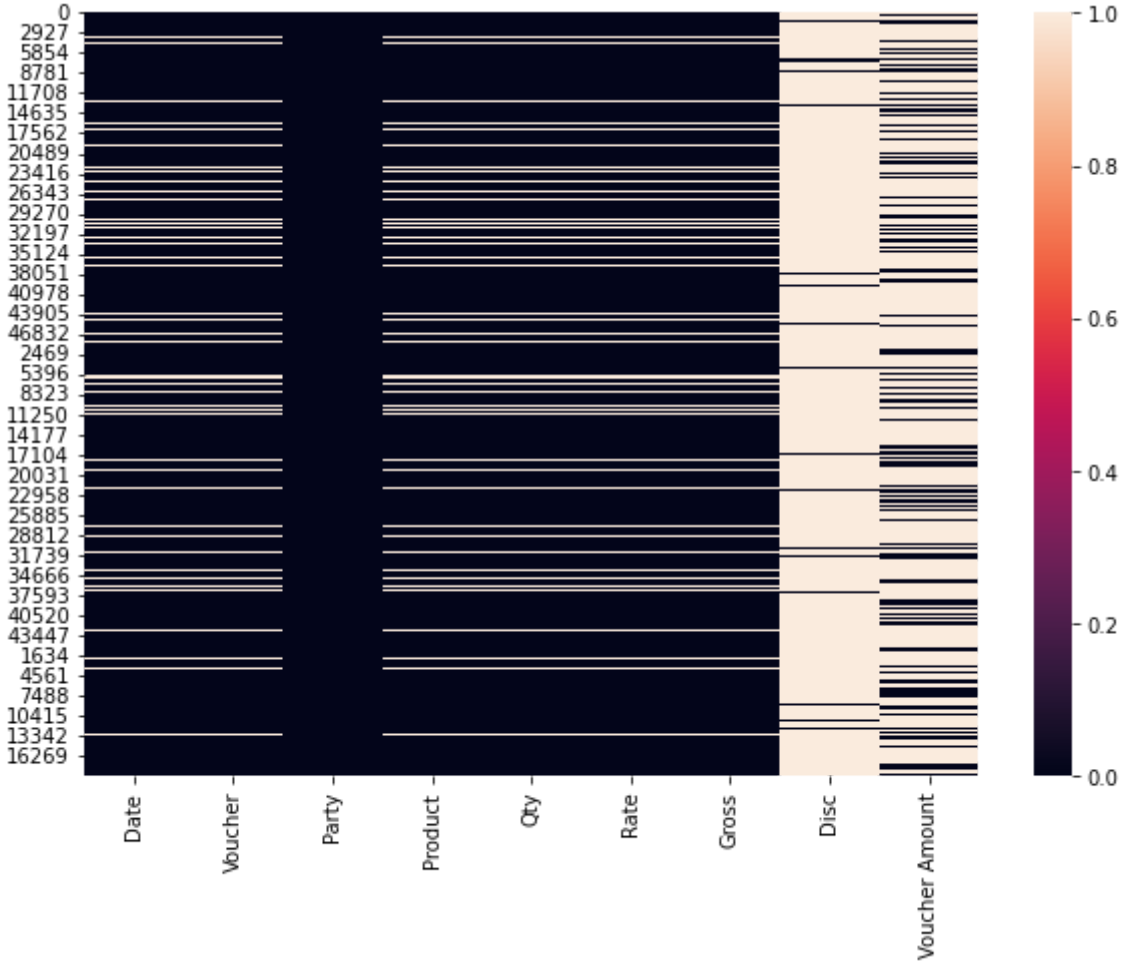
```
import seaborn as sns
```

In [18]:

```
plt.figure(figsize=(10,7))
sns.heatmap(sales_complete_data.isnull())
```

Out[18]:

<AxesSubplot:>



from the above heatmap we can see that 'disc'& 'voucher amount' are having more null values

In [19]:

```
data_1=sales_2017.drop(['Disc','Voucher Amount'], axis= 1)
data_1
```

Out[19]:

	Date	Voucher	Party	Product	Qty	Rate	Gr
0	1/4/2017	Sal:1	SOLANKI PLASTICS	DONA-VAI-9100	2	1,690.00	3,38(
1	1/4/2017	Sal:1	SOLANKI PLASTICS	LITE FOAM(1200)	6	1,620.00	9,72(
2	1/4/2017	Sal:2	SARNESWARA TRADERS	VISHNU CHOTA WINE	500	23	11,50(
3	1/4/2017	Sal:2	SARNESWARA TRADERS	LITE FOAM(1200)	6	1,620.00	9,72(
4	1/4/2017	Sal:2	SARNESWARA TRADERS	DONA-VAI-9100	5	1,690.00	8,45(
...	...	...	...	...	...	...	
47285	31/03/2018	Sal:10042	Vkp	10*10 SHEET	25	137	3,42
47286	NaN	NaN	NaN	NaN	NaN	NaN	N
47287	NaN	NaN	NaN	NaN	NaN	NaN	N
47288	NaN	Total	NaN	NaN	607,734.60	669,300.49	9,953,81(
47289	NaN	Total	NaN	NaN	7,593,062.00	8,309,116.00	115,778,72

47290 rows × 7 columns

In [20]:

```
data_1.isnull().sum()
```

Out[20]:

```
Date       5388
Voucher    5375
Party        15
Product    5388
Qty        5375
Rate       5376
Gross      5376
dtype: int64
```

In [21]:

```
data_1.dtypes
```

Out[21]:

```
Date       object
Voucher    object
Party      object
Product    object
Qty        object
Rate       object
Gross      object
dtype: object
```

In [33]:

```
data_1['Date']=pd.to_datetime(data_1['Date'])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\anaconda3\lib\site-packages\pandas\core\arrays\datetimes.py in objects_to_
datetime64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, allow_
object, allow_mixed)
   2186         try:
-> 2187             values, tz_parsed = conversion.datetime_to_datetime64(da
ta.ravel("K"))
   2188             # If tzaware, these values represent unix timestamps, so
we

~\anaconda3\lib\site-packages\pandas\_libs\tslibs\conversion.pyx in pandas._
libs.tslibs.conversion.datetime_to_datetime64()

TypeError: Unrecognized value type: <class 'str'>

During handling of the above exception, another exception occurred:

ParserError                               Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_9376/1932038014.py in <module>
----> 1 data_1['Date']=pd.to_datetime(data_1['Date'])

~\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py in to_datetime
(arg, errors, dayfirst, yearfirst, utc, format, exact, unit, infer_datetime_
format, origin, cache)
    881                 result = result.tz_localize(tz)  # type: ignore[call
-arg]
    882     elif isinstance(arg, ABCSeries):
--> 883         cache_array = _maybe_cache(arg, format, cache, convert_listl
ike)
    884         if not cache_array.empty:
    885             result = arg.map(cache_array)

~\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py in _maybe_cache
(arg, format, cache, convert_listlike)
    193         unique_dates = unique(arg)
    194         if len(unique_dates) < len(arg):
--> 195             cache_dates = convert_listlike(unique_dates, format)
    196             cache_array = Series(cache_dates, index=unique_dates)
    197             # GH#39882 and GH#35888 in case of None and NaT we get d
uplicates

~\anaconda3\lib\site-packages\pandas\core\tools\datetimes.py in _convert_lis
tlike_datetimes(arg, format, name, tz, unit, errors, infer_datetime_format,
 dayfirst, yearfirst, exact)
    399     assert format is None or infer_datetime_format
    400     utc = tz == "utc"
--> 401     result, tz_parsed = objects_to_datetime64ns(
    402         arg,
    403         dayfirst=dayfirst,

~\anaconda3\lib\site-packages\pandas\core\arrays\datetimes.py in objects_to_
datetime64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, allow_
object, allow_mixed)
   2191             return values.view("i8"), tz_parsed
   2192         except (ValueError, TypeError):
-> 2193             raise err
```

```
   2194
   2195     if tz_parsed is not None:

~\anaconda3\lib\site-packages\pandas\core\arrays\datetimes.py in objects_to_
datetime64ns(data, dayfirst, yearfirst, utc, errors, require_iso8601, allow_
object, allow_mixed)
   2173     order: Literal["F", "C"] = "F" if flags.f_contiguous else "C"
   2174     try:
-> 2175         result, tz_parsed = tslib.array_to_datetime(
   2176             data.ravel("K"),
   2177             errors=errors,

~\anaconda3\lib\site-packages\pandas\_libs\tslib.pyx in pandas._libs.tslib.a
rray_to_datetime()

~\anaconda3\lib\site-packages\pandas\_libs\tslib.pyx in pandas._libs.tslib.a
rray_to_datetime()

~\anaconda3\lib\site-packages\pandas\_libs\tslib.pyx in pandas._libs.tslib._
array_to_datetime_object()

~\anaconda3\lib\site-packages\pandas\_libs\tslib.pyx in pandas._libs.tslib._
array_to_datetime_object()

~\anaconda3\lib\site-packages\pandas\_libs\tslibs\parsing.pyx in pandas._lib
s.tslibs.parsing.parse_datetime_string()

~\anaconda3\lib\site-packages\dateutil\parser\_parser.py in parse(timestr, p
arserinfo, **kwargs)
   1366         return parser(parserinfo).parse(timestr, **kwargs)
   1367     else:
-> 1368         return DEFAULTPARSER.parse(timestr, **kwargs)
   1369
   1370

~\anaconda3\lib\site-packages\dateutil\parser\_parser.py in parse(self, time
str, default, ignoretz, tzinfos, **kwargs)
    644
    645         if len(res) == 0:
--> 646             raise ParserError("String does not contain a date: %s",
 timestr)
    647
    648         try:

ParserError: String does not contain a date:
```

In [22]:

```
data_1['Rate']=data_1['Rate'].astype(int)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_9376/1898249065.py in <module>
----> 1 data_1['Rate']=data_1['Rate'].astype(int)

~\anaconda3\lib\site-packages\pandas\core\generic.py in astype(self, dtype, copy, errors)
   5813         else:
   5814             # else, only a single dtype is given
-> 5815             new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)
   5816             return self._constructor(new_data).__finalize__(self, method="astype")
   5817

~\anaconda3\lib\site-packages\pandas\core\internals\managers.py in astype(self, dtype, copy, errors)
    416
    417     def astype(self: T, dtype, copy: bool = False, errors: str = "raise") -> T:
--> 418         return self.apply("astype", dtype=dtype, copy=copy, errors=errors)
    419
    420     def convert(

~\anaconda3\lib\site-packages\pandas\core\internals\managers.py in apply(self, f, align_keys, ignore_failures, **kwargs)
    325                     applied = b.apply(f, **kwargs)
    326                 else:
--> 327                     applied = getattr(b, f)(**kwargs)
    328             except (TypeError, NotImplementedError):
    329                 if not ignore_failures:

~\anaconda3\lib\site-packages\pandas\core\internals\blocks.py in astype(self, dtype, copy, errors)
    589         values = self.values
    590
--> 591         new_values = astype_array_safe(values, dtype, copy=copy, errors=errors)
    592
    593         new_values = maybe_coerce_values(new_values)

~\anaconda3\lib\site-packages\pandas\core\dtypes\cast.py in astype_array_safe(values, dtype, copy, errors)
   1307
   1308     try:
-> 1309         new_values = astype_array(values, dtype, copy=copy)
   1310     except (ValueError, TypeError):
   1311         # e.g. astype_nansafe can fail on object-dtype of strings

~\anaconda3\lib\site-packages\pandas\core\dtypes\cast.py in astype_array(values, dtype, copy)
   1255
   1256     else:
-> 1257         values = astype_nansafe(values, dtype, copy=copy)
```

```
   1258
   1259     # in pandas we don't store numpy str dtypes, so convert to obj
ect

~\anaconda3\lib\site-packages\pandas\core\dtypes\cast.py in astype_nansafe
(arr, dtype, copy, skipna)
   1172         # work around NumPy brokenness, #1987
   1173         if np.issubdtype(dtype.type, np.integer):
-> 1174             return lib.astype_intsafe(arr, dtype)
   1175
   1176         # if we have a datetime/timedelta array of objects

~\anaconda3\lib\site-packages\pandas\_libs\lib.pyx in pandas._libs.lib.ast
ype_intsafe()

ValueError: invalid literal for int() with base 10: '1,690.00'
```

In [23]:

```
data_2=sales_2018.drop(['Disc','Voucher Amount'], axis= 1)
data_2
```

Out[23]:

	Date	Voucher	Party	Product	Qty	Rate	Gross
0	1/4/2018	Sal:146	TP13	SILVER POUCH 9*12	50	85	4,250.00
1	1/4/2018	Sal:146	TP13	RUBBER	5	290	1,450.00
2	1/4/2018	Sal:146	TP13	DURGA 10*12 Blue	1,600.00	5.5	8,800.00
3	1/4/2018	Sal:146	TP13	DURGA 13*16 BLUE	400	11	4,400.00
4	1/4/2018	Sal:146	TP13	10*12 SARAS-NAT	600	8.1	4,860.00
...	...	...	...	...	...	...	...
44735	31/03/2019	Sal:9610	HAMPI FOODS	SPOON SOOFY	200	40	8,000.00
44736	NaN	NaN	NaN	NaN	NaN	NaN	NaN
44737	NaN	NaN	NaN	NaN	NaN	NaN	NaN
44738	NaN	Total	NaN	NaN	666,056.00	1,067,808.80	10,796,991.30
44739	NaN	Total	NaN	NaN	7,097,803.00	10,024,197.00	117,897,671.80

44740 rows × 7 columns

In [24]:

```
data_3=sales_2019.drop(['Disc','Voucher Amount'], axis= 1)
data_3
```

Out[24]:

	Date	Voucher	Party	Product	Qty	Rate	Gross
0	1/4/2019	Sal:687	BALAJI PLASTICS	DONA-VAI-9100	1	1,730.00	1,730.00
1	1/4/2019	Sal:687	BALAJI PLASTICS	SMART BOUL(48)	1	1,730.00	1,730.00
2	1/4/2019	Sal:688	BALAJI PLASTICS	Vishnu Ice	110	18.5	2,035.00
3			28/3		0	0	
4	1/4/2019	Sal:689	BALAJI PLASTICS	100LEAF - SP	3	585	1,755.00
...	...	...	...	...	...	...	...
19171	10/10/2019	Sal:4935	K.SRIHARI	13*16 WHITE RK	400	16	6,400.00
19172	NaN	NaN	NaN	NaN	NaN	NaN	NaN
19173	NaN	NaN	NaN	NaN	NaN	NaN	NaN
19174	NaN	Total	NaN	NaN	99,284.90	175,381.65	2,203,649.50
19175	NaN	Total	NaN	NaN	2,710,193.00	5,519,888.40	53,360,791.40

19176 rows × 7 columns

In [25]:

```
my_cleaned_data=pd.concat([data_1,data_2,data_3])
my_cleaned_data
```

Out[25]:

	Date	Voucher	Party	Product	Qty	Rate	Gros
0	1/4/2017	Sal:1	SOLANKI PLASTICS	DONA-VAI-9100	2	1,690.00	3,380.0
1	1/4/2017	Sal:1	SOLANKI PLASTICS	LITE FOAM(1200)	6	1,620.00	9,720.0
2	1/4/2017	Sal:2	SARNESWARA TRADERS	VISHNU CHOTA WINE	500	23	11,500.0
3	1/4/2017	Sal:2	SARNESWARA TRADERS	LITE FOAM(1200)	6	1,620.00	9,720.0
4	1/4/2017	Sal:2	SARNESWARA TRADERS	DONA-VAI-9100	5	1,690.00	8,450.0
...	...	...	...	...	...	...	
19171	10/10/2019	Sal:4935	K.SRIHARI	13*16 WHITE RK	400	16	6,400.0
19172	NaN	NaN	NaN	NaN	NaN	NaN	Na
19173	NaN	NaN	NaN	NaN	NaN	NaN	Na
19174	NaN	Total	NaN	NaN	99,284.90	175,381.65	2,203,649.5
19175	NaN	Total	NaN	NaN	2,710,193.00	5,519,888.40	53,360,791.4

111206 rows × 7 columns

In [26]:

```
my_cleaned_data.isnull().sum()
```

Out[26]:

```
Date       12591
Voucher    12557
Party         40
Product    12591
Qty        12557
Rate       12558
Gross      12558
dtype: int64
```

In [27]:

```
my_cleaned_data.dropna(how='all',inplace=True)
```

In [28]:

```
my_cleaned_data.isnull().sum()
```

Out[28]:

```
Date       12585
Voucher    12551
Party         34
Product    12585
Qty        12551
Rate       12552
Gross      12552
dtype: int64
```

In [29]:

```
my_cleaned_data=my_cleaned_data[my_cleaned_data['Product'].notna()]
```

In [30]:

```
my_cleaned_data.isnull().sum()
```

Out[30]:

```
Date       0
Voucher    0
Party      0
Product    0
Qty        0
Rate       1
Gross      1
dtype: int64
```

In [31]:

```
my_cleaned_data.dtypes
```

Out[31]:

```
Date       object
Voucher    object
Party      object
Product    object
Qty        object
Rate       object
Gross      object
dtype: object
```

In [32]:

```
my_cleaned_data['Rate ']=my_cleaned_data['Rate'].astype(int)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_9376/2031643881.py in <module>
----> 1 my_cleaned_data['Rate ']=my_cleaned_data['Rate'].astype(int)

~\anaconda3\lib\site-packages\pandas\core\generic.py in astype(self, dtype, copy, errors)
   5813         else:
   5814             # else, only a single dtype is given
-> 5815             new_data = self._mgr.astype(dtype=dtype, copy=copy, errors=errors)
   5816             return self._constructor(new_data).__finalize__(self, method="astype")
   5817

~\anaconda3\lib\site-packages\pandas\core\internals\managers.py in astype(self, dtype, copy, errors)
    416
    417     def astype(self: T, dtype, copy: bool = False, errors: str = "raise") -> T:
--> 418         return self.apply("astype", dtype=dtype, copy=copy, errors=errors)
    419
    420     def convert(

~\anaconda3\lib\site-packages\pandas\core\internals\managers.py in apply(self, f, align_keys, ignore_failures, **kwargs)
    325                     applied = b.apply(f, **kwargs)
    326                 else:
--> 327                     applied = getattr(b, f)(**kwargs)
    328             except (TypeError, NotImplementedError):
    329                 if not ignore_failures:

~\anaconda3\lib\site-packages\pandas\core\internals\blocks.py in astype(self, dtype, copy, errors)
    589         values = self.values
    590
--> 591         new_values = astype_array_safe(values, dtype, copy=copy, errors=errors)
    592
    593         new_values = maybe_coerce_values(new_values)

~\anaconda3\lib\site-packages\pandas\core\dtypes\cast.py in astype_array_safe(values, dtype, copy, errors)
   1307
   1308     try:
-> 1309         new_values = astype_array(values, dtype, copy=copy)
   1310     except (ValueError, TypeError):
   1311         # e.g. astype_nansafe can fail on object-dtype of strings

~\anaconda3\lib\site-packages\pandas\core\dtypes\cast.py in astype_array(values, dtype, copy)
   1255
   1256     else:
-> 1257         values = astype_nansafe(values, dtype, copy=copy)
```

```
   1258
   1259     # in pandas we don't store numpy str dtypes, so convert to obj
ect

~\anaconda3\lib\site-packages\pandas\core\dtypes\cast.py in astype_nansafe
(arr, dtype, copy, skipna)
   1172         # work around NumPy brokenness, #1987
   1173         if np.issubdtype(dtype.type, np.integer):
-> 1174             return lib.astype_intsafe(arr, dtype)
   1175
   1176         # if we have a datetime/timedelta array of objects

~\anaconda3\lib\site-packages\pandas\_libs\lib.pyx in pandas._libs.lib.ast
ype_intsafe()

ValueError: invalid literal for int() with base 10: '1,690.00'
```

In [ ]:

In [ ]: