```
import pandas as pd
```

1. Transactions Sheet Analysis

```
Sheet1 = pd.read_excel("Raw_Data.xlsx", sheet_name='Transactions')

#Checking top rows for analysis
Sheet1.head()
```

	transaction_id	product_id	customer_id	transaction_date	online_order	o
0	1	2	2950	2017-02-25	0.0	
1	2	3	3120	2017-05-21	1.0	
2	3	37	402	2017-10-16	0.0	
3	4	88	3135	2017-08-31	0.0	
4	5	78	787	2017-10-01	1.0	

```
#Getting Total Rows and Columns Information
Sheet1.shape
```

```
(20000, 13)
```

```
#Information about Data types used, Total columns filled etc.
Sheet1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 13 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   transaction_id           20000 non-null  int64
 1   product_id               20000 non-null  int64
 2   customer_id              20000 non-null  int64
 3   transaction_date         20000 non-null  datetime64[ns]
 4   online_order             19640 non-null  float64
 5   order_status             20000 non-null  object
 6   brand                    19803 non-null  object
 7   product_line             19803 non-null  object
 8   product_class            19803 non-null  object
 9   product_size             19803 non-null  object
 10  list_price               20000 non-null  float64
 11  standard_cost            19803 non-null  float64
 12  product_first_sold_date  19803 non-null  float64
dtypes: datetime64[ns](1), float64(4), int64(3), object(5)
memory usage: 2.0+ MB
```

Here Some fields columns are not filled completely and they are assumed to be blank. The blank rows or data needs to be cleaned before further analysis.

Columns which contains blank, null data are:

1. online_order
2. brand
3. product_line
4. product_class
5. product_size
6. standard_cost
7. product_first_sold_date

```
#How many values in each column are missing?
Sheet1.isnull().sum()
```

```
transaction_id             0
product_id                 0
customer_id                0
transaction_date           0
```

```
online_order              360
order_status                0
brand                     197
product_line              197
product_class             197
product_size              197
list_price                  0
standard_cost             197
product_first_sold_date   197
dtype: int64
```

The above analysis shows that there are so many values missing in the above particular columns

```
#Searching for Duplicate Values in Transaction Sheet
Sheet1.duplicated().sum()
```

```
0
```

The above analysis shows that there are no duplicate values in the Transaction sheet which is a good thing

```
#Searching for uniqueness
Sheet1.nunique()
```

```
transaction_id             20000
product_id                   101
customer_id                 3494
transaction_date             364
online_order                   2
order_status                   2
brand                          6
product_line                   4
product_class                  3
product_size                   3
list_price                   296
standard_cost                103
product_first_sold_date      100
dtype: int64
```

The above analysis suggests that there are 20k transaction_id which are totally unique and hence we can say that each row or record can be uniquely identified using transaction_id

2. Customer Demographic Sheet Analysis

```
Sheet2 = pd.read_excel("Raw_Data.xlsx", sheet_name='CustomerDemographic')

#Checking some rows to get more info
Sheet2.head()
```

```
<ipython-input-28-55b042292ad5>:1: FutureWarning: Inferring datetime64[ns] from data containing strings is
  Sheet2 = pd.read_excel("Raw_Data.xlsx", sheet_name='CustomerDemographic')
```

	customer_id	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB
0	1	Laraine	Medendorp	F	93	1953-10-12
1	2	Eli	Bockman	Male	81	1980-12-16
2	3	Arlin	Dearle	Male	61	1954-01-20
3	4	Talbot	NaN	Male	33	1961-10-03
4	5	Sheila-kathryn	Calton	Female	56	1977-05-13

```
#Total Rows and Columns Information
Sheet2.shape
```

```
(4000, 13)
```

It seems like there are problems in the data related to the date of birth (DOB Column)

The date format also contains string format data which may cause problem and it needs to be converted for further analysis

```
#Information about Data types used, Total columns filled etc.
Sheet2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4000 entries, 0 to 3999
Data columns (total 13 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   customer_id                          4000 non-null   int64
 1   first_name                           4000 non-null   object
 2   last_name                            3875 non-null   object
 3   gender                               4000 non-null   object
 4   past_3_years_bike_related_purchases  4000 non-null   int64
 5   DOB                                  3913 non-null   datetime64[ns]
 6   job_title                            3494 non-null   object
 7   job_industry_category                3344 non-null   object
 8   wealth_segment                       4000 non-null   object
 9   deceased_indicator                   4000 non-null   object
 10  default                              3698 non-null   object
 11  owns_car                             4000 non-null   object
 12  tenure                               3913 non-null   float64
dtypes: datetime64[ns](1), float64(1), int64(2), object(9)
memory usage: 406.4+ KB
```

```
#Total Empty cells or fields in this sheet
Sheet2.isnull().sum()
```

```
customer_id                              0
first_name                               0
last_name                              125
gender                                   0
past_3_years_bike_related_purchases      0
DOB                                     87
job_title                              506
job_industry_category                  656
wealth_segment                           0
deceased_indicator                       0
default                                302
owns_car                                 0
tenure                                  87
dtype: int64
```

From this analysis we get to know that there are some fields or cells which are blank

These blank values can cause errors in calculations, the columns which contains blank values are:

1. last_name
2. DOB
3. job_title
4. job_industry_category
5. default
6. tenure

```
#Finding Duplicate Values
Sheet2.duplicated().sum()
```

```
0
```

There are no duplicated values found in this sheet of Customer Demographics

```
#Searching for uniqueness
Sheet2.nunique()
```

```
customer_id                            4000
first_name                             3139
last_name                              3725
gender                                    6
past_3_years_bike_related_purchases     100
DOB                                    3448
job_title                               195
job_industry_category                     9
wealth_segment                            3
deceased_indicator                        2
default                                  90
owns_car                                  2
tenure                                   22
dtype: int64
```

The above analysis shows that customer_id can be used as a unique key to identify the records

```
#Checking the columns information (Seemed to contain various unknown values)
Sheet2['default'].value_counts()
```

```
100                                   113
1                                    112
-1                                   111
-100                                   99
ÙiÙ¢Ù£                                   53
                                    ...
testâ testâ«                             31
/dev/null; touch /tmp/blns.fail ; echo    30
âªâªtestâª                               29
ì¸ëë°í ë¥´                              27
,ãã»:*:ã»ãâ( â» Ï â» )ãã»:*:ã»ãâ          25
Name: default, Length: 90, dtype: int64
```

```
#Checking the columns information (Seemed to contain various different values)
Sheet2['gender'].value_counts()
```

```
Female    2037
Male      1872
U           88
F            1
Femal        1
M            1
Name: gender, dtype: int64
```

The above two analysis shows that there are useless values in the default and gender field

The default column contains various values which cannot be understood and doesn't seem to be related to any data and hence it can be dropped.

The gender column contains various types of representations which needs to be merged together for making the analysis easier Male - M, Female - F etc. Also there seems to be 'Femal' which is an error. Either all values can be represented using (Male-Female-Unidentified) or (M-F-U)

3. New Customer List Sheet Analysis

```
Sheet3 = pd.read_excel("Raw_Data.xlsx", sheet_name='NewCustomerList')

#Checking few rows for general information
Sheet3.head()
```

```
<ipython-input-7-493d987b897d>:1: FutureWarning: Inferring datetime64[ns] from data containing strings is
  Sheet3 = pd.read_excel("Raw_Data.xlsx", sheet_name='NewCustomerList')
```

	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job_title
0	Chickie	Brister	Male	86	1957-07-12	General Manager
1	Morly	Genery	Male	69	1970-03-22	Structural Engineer
2	Ardelis	Forrester	Female	10	1974-08-28	Senior Cost Accountant
3	Lucine	Stutt	Female	64	1979-01-28	Account Representative III
4	Melinda	Hadlee	Female	34	1965-09-21	Financial Analyst

5 rows × 23 columns

```
#Total rows and columns in this sheet
Sheet3.shape
```

```
(1000, 23)
```

```
#Data type and total values filled information
Sheet3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 23 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   first_name                           1000 non-null   object
 1   last_name                            971 non-null    object
 2   gender                               1000 non-null   object
 3   past_3_years_bike_related_purchases  1000 non-null   int64
 4   DOB                                  983 non-null    datetime64[ns]
 5   job_title                            894 non-null    object
 6   job_industry_category                835 non-null    object
 7   wealth_segment                       1000 non-null   object
 8   deceased_indicator                   1000 non-null   object
 9   owns_car                             1000 non-null   object
 10  tenure                               1000 non-null   int64
 11  address                              1000 non-null   object
 12  postcode                             1000 non-null   int64
 13  state                                1000 non-null   object
 14  country                              1000 non-null   object
 15  property_valuation                   1000 non-null   int64
 16  Unnamed: 16                          1000 non-null   float64
 17  Unnamed: 17                          1000 non-null   float64
 18  Unnamed: 18                          1000 non-null   float64
 19  Unnamed: 19                          1000 non-null   float64
 20  Unnamed: 20                          1000 non-null   int64
 21  Rank                                 1000 non-null   int64
 22  Value                                1000 non-null   float64
dtypes: datetime64[ns](1), float64(5), int64(6), object(11)
memory usage: 179.8+ KB
```

This analysis shows us that there are few unnecessary columns

Some unnecessary columns are 'Unnamed' from the range 16 to 20. These needs to be dropped for further analysis

We can also see that there are a few columns which contains empty or blank data

```
#Dropping unnecessary columns for further analysis
NewSheet3 = Sheet3.drop(['Unnamed: 16','Unnamed: 17','Unnamed: 18','Unnamed: 19','Unnamed: 20'], axis=1)

#Checking Head for new sheet
NewSheet3.head()
```

	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job_title
0	Chickie	Brister	Male	86	1957-07-12	General Manager
1	Morly	Genery	Male	69	1970-03-22	Structural Engineer
2	Ardelis	Forrester	Female	10	1974-08-28	Senior Cost Accountant
3	Lucine	Stutt	Female	64	1979-01-28	Account Representative III
4	Melinda	Hadlee	Female	34	1965-09-21	Financial Analyst

```
#Checking for blank or empty values
NewSheet3.isnull().sum()
```

```
first_name                               0
last_name                               29
gender                                   0
past_3_years_bike_related_purchases      0
DOB                                     17
job_title                              106
job_industry_category                  165
wealth_segment                           0
deceased_indicator                       0
owns_car                                 0
tenure                                   0
```

```
address                    0
postcode                   0
state                      0
country                    0
property_valuation         0
Rank                       0
Value                      0
dtype: int64
```

The above analysis shows that there are few columns which are blank and these rows needs to be removed

The columns which contains null values are:

1. last_name
2. DOB
3. job_title
4. job_industry_category

```
#Checking Duplicate values
NewSheet3.duplicated().sum()
```

```
0
```

There seems to be no duplicate records in this sheet as well

```
#Checking Uniqueness
NewSheet3.nunique()
```

```
first_name                             940
last_name                              961
gender                                   3
past_3_years_bike_related_purchases    100
DOB                                    958
job_title                              184
job_industry_category                    9
wealth_segment                           3
deceased_indicator                       1
owns_car                                 2
tenure                                  23
address                               1000
postcode                               522
state                                    3
country                                  1
property_valuation                      12
Rank                                   324
Value                                  324
dtype: int64
```

According to the total number of rows in the sheet address seems to be the only unique key in this New Customers List

Address doesn't seem fit for this role and it can be changed. Maybe a randomly generated string can be used to uniquely identify each record

```
#Checking genders in New Customer List
NewSheet3['gender'].value_counts()
```

```
Female    513
Male      470
U          17
Name: gender, dtype: int64
```

This seems fine but replacing U with Unidentified will make the records look more sorted and easier to read

4. Customer Address Sheet Analysis

```
Sheet4 = pd.read_excel("Raw_Data.xlsx", sheet_name='CustomerAddress')

#Reading top 5 rows
Sheet4.head()
```

	Note: The data and information in this document is reflective of a hypothetical situation and client. This document is to be used for KPMG Virtual Internship purposes only.	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5
0	customer_id	address	postcode	state	country	property_valuation
1	1	060 Morning Avenue	2016	New South Wales	Australia	10

```
#Getting total rows and columns
Sheet4.shape
```

```
(3999, 6)
```

```
#Understanding data types and filled columns
Sheet4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3999 entries, 0 to 3998
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   customer_id         3999 non-null   int64
 1   address             3999 non-null   object
 2   postcode            3999 non-null   int64
 3   state               3999 non-null   object
 4   country             3999 non-null   object
 5   property_valuation  3999 non-null   int64
dtypes: int64(3), object(3)
memory usage: 187.6+ KB
```

```
#Checking for duplicate values
Sheet4.duplicated().sum()
```

```
0
```

```
#Checking consistency of Country
Sheet4['country'].value_counts()
```

```
Australia    3999
Name: country, dtype: int64
```

```
#Checking consistency of States
Sheet4['state'].value_counts()
```

```
NSW                2054
VIC                 939
QLD                 838
New South Wales      86
Victoria             82
Name: state, dtype: int64
```

From the above analysis it is safe to say that the Address table is the most consistent among others

There are no duplicate or null values encountered in this table and is good to go.

The country and states also seems to be in order and any duplication in form of different name is not found.

CONCLUSION

The above data have been successfully analyzed and below are some findings

Sheet 1: Transactions

1. Some values in particular columns seems to be blank
2. No duplicate values encountered
3. Transaction_id is unique and can be used to identify records.

Sheet 2: Customer Demographic

1. DOB column seems to contain strings which should be dates instead
2. Some cells are blank

3. No duplicate values encountered
4. A column named 'default' seems to contain unknown information which need to be dropped
5. Genders are represented using various methods eg(Male, M, Female, F) which needs to be changed to specific style

Sheet 3: New Customers List

1. There are a few unnecessary columns which are 'unnamed'
2. There seems to be empty cells in few columns
3. There is no duplication recorded
4. The U in gender can be replaced as Unidentified to make it match with Male-Female format

Sheet 4: Customer Address

1. No empty rows or cells
2. No duplications
3. All the data is unique and consistent/accurate