Task 1: Data Quality Assesment

Assessment of data quality and completeness in preparation for analysis

In [1]:
```
# Importing Libraries
import numpy as np
import pandas as pd
```

In [2]:
```
# Reading the file
df = pd.ExcelFile("KPMGTask1.xlsx")
```

In [3]:
```
# Reading Each File Seperately
Transactions = pd.read_excel(df, "Transactions")
NewCustomerList = pd.read_excel(df, "NewCustomerList")
CustomerDemographic = pd.read_excel(df, "CustomerDemographic")
CustomerAddress = pd.read_excel(df, "CustomerAddress")
```

1. Transactions

In [4]:
```
Transactions.head()
```

Out[4]:

	transaction_id	product_id	customer_id	transaction_date	online_order	order_status	brand	product_line	product_class	product_size	list_price	standard_
0	1	2	2950	42791	0.0	Approved	Solex	Standard	medium	medium	71.49	
1	2	3	3120	42876	1.0	Approved	Trek Bicycles	Standard	medium	large	2091.47	3
2	3	37	402	43024	0.0	Approved	OHM Cycles	Standard	low	medium	1793.43	2
3	4	88	3135	42978	0.0	Approved	Norco Bicycles	Standard	medium	medium	1198.46	3
4	5	78	787	43009	1.0	Approved	Giant Bicycles	Standard	medium	large	1765.30	7

In [5]:
```
Transactions.dtypes
```

Out[5]:
```
transaction_id               int64
product_id                   int64
customer_id                  int64
transaction_date             int64
online_order               float64
order_status                object
brand                       object
product_line                object
product_class               object
product_size                object
list_price                 float64
standard_cost              float64
product_first_sold_date    float64
dtype: object
```

In [6]:
```
#Using the required columns
Transactions = Transactions.iloc[:, 0:13]
Transactions.head()
```

Out[6]:

	transaction_id	product_id	customer_id	transaction_date	online_order	order_status	brand	product_line	product_class	product_size	list_price	standard_
0	1	2	2950	42791	0.0	Approved	Solex	Standard	medium	medium	71.49	
1	2	3	3120	42876	1.0	Approved	Trek Bicycles	Standard	medium	large	2091.47	3
2	3	37	402	43024	0.0	Approved	OHM Cycles	Standard	low	medium	1793.43	2
3	4	88	3135	42978	0.0	Approved	Norco Bicycles	Standard	medium	medium	1198.46	3
4	5	78	787	43009	1.0	Approved	Giant Bicycles	Standard	medium	large	1765.30	7

In [7]:
```
Transactions.shape
```

Out[7]:
```
(20000, 13)
```

In [8]:
```
# Checking Null Values
Transactions.isnull().sum()
```

Out[8]:
```
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
```

There are 7 Null Columns present in the Transactions file.

In [9]:
```
# Checking Duplicates Value
Transactions.duplicated().sum()
```

Out[9]:
```
0
```

There are No Duplicates Vlaue Present in the data.

In [10]:
```
# Checking unique
Transactions.nunique()
```

Out[10]:
```
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
```

In [11]:
```
# Exploring Columns
Transactions.columns
```

Out[11]:
```
Index(['transaction_id', 'product_id', 'customer_id', 'transaction_date',
       'online_order', 'order_status', 'brand', 'product_line',
       'product_class', 'product_size', 'list_price', 'standard_cost',
       'product_first_sold_date'],
      dtype='object')
```

In [12]:
```
Transactions['order_status'].value_counts()
```

Out[12]:
```
Approved     19821
Cancelled      179
Name: order_status, dtype: int64
```

In [13]:
```
Transactions['brand'].value_counts()
```

Out[13]:
```
Solex             4253
Giant Bicycles    3312
WeareA2B          3295
OHM Cycles        3043
Trek Bicycles     2990
Norco Bicycles    2910
Name: brand, dtype: int64
```

In [14]:
```
Transactions['product_line'].value_counts()
```

Out[14]:
```
Standard    14176
Road         3970
Touring      1234
Mountain      423
Name: product_line, dtype: int64
```

In [15]:
```
Transactions['product_class'].value_counts()
```

Out[15]:
```
medium    13826
high       3013
low        2964
Name: product_class, dtype: int64
```

In [16]:
```
Transactions['product_size'].value_counts()
```

Out[16]:
```
medium    12990
large      3976
small      2837
Name: product_size, dtype: int64
```

In [17]:
```
Transactions['product_first_sold_date']
```

Out[17]:
```
0        41245.0
1        41701.0
2        36361.0
3        36145.0
4        42226.0
          ...
19995    37823.0
19996    35560.0
19997    40410.0
19998    38216.0
19999    36334.0
Name: product_first_sold_date, Length: 20000, dtype: float64
```

Above Column is not in DATE/TIME Format, so we have to convert it into DATE/TIME Format.

In [18]:
```
# Converting integer to date-time
Transactions['product_first_sold_date'] = pd.to_datetime(Transactions['product_first_sold_date'], unit='s')
Transactions['product_first_sold_date'].head()
```

Out[18]:
```
0   1970-01-01 11:27:25
1   1970-01-01 11:35:01
2   1970-01-01 10:06:01
3   1970-01-01 10:02:25
4   1970-01-01 11:43:46
Name: product_first_sold_date, dtype: datetime64[ns]
```

In [19]:
```
Transactions['product_first_sold_date'].head(10)
```

Out[19]:
```
0   1970-01-01 11:27:25
1   1970-01-01 11:35:01
2   1970-01-01 10:06:01
3   1970-01-01 10:02:25
4   1970-01-01 11:43:46
5   1970-01-01 10:50:31
6   1970-01-01 09:29:25
7   1970-01-01 11:05:15
8   1970-01-01 09:17:35
9   1970-01-01 10:36:56
Name: product_first_sold_date, dtype: datetime64[ns]
```

The values in the product_first_sold_date column is not correct. It is showing everything happening at the same day but at different times.

2. New Customer

In [20]:
```
NewCustomerList.head()
```

Out[20]:

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

5 rows × 23 columns

In [21]:
```
NewCustomerList.dtypes
```

Out[21]:
```
first_name                           object
last_name                            object
gender                               object
past_3_years_bike_related_purchases   int64
DOB                          datetime64[ns]
job_title                            object
job_industry_category                object
wealth_segment                       object
deceased_indicator                   object
owns_car                             object
tenure                                int64
address                              object
postcode                              int64
state                                object
country                              object
property_valuation                    int64
Unnamed: 16                         float64
Unnamed: 17                         float64
Unnamed: 18                         float64
Unnamed: 19                         float64
Unnamed: 20                           int64
Rank                                  int64
Value                               float64
dtype: object
```

In [22]:
```
#Dropping the unnamed columns
NewCustomerList.drop(['Unnamed: 16', 'Unnamed: 17', 'Unnamed: 18',
       'Unnamed: 19', 'Unnamed: 20'], axis=1, inplace=True)
```

In [23]:
```
NewCustomerList.shape
```

Out[23]:
```
(1000, 18)
```

In [24]:
```
#Checking for null values
NewCustomerList.isnull().sum()
```

Out[24]:
```
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
```

In [25]:
```
#Checking for duplicate values
NewCustomerList.duplicated().sum()
```

Out[25]:
```
0
```

In [26]:
```
#Checking for uniquess of each column
NewCustomerList.nunique()
```

Out[26]:
```
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                               319
dtype: int64
```

In [27]:
```
# Exploring Column
NewCustomerList.columns
```

Out[27]:
```
Index(['first_name', 'last_name', 'gender',
       'past_3_years_bike_related_purchases', 'DOB', 'job_title',
       'job_industry_category', 'wealth_segment', 'deceased_indicator',
       'owns_car', 'tenure', 'address', 'postcode', 'state', 'country',
       'property_valuation', 'Rank', 'Value'],
      dtype='object')
```

In [28]:
```
NewCustomerList['gender'].value_counts()
```

Out[28]:
```
Female    513
Male      470
U          17
Name: gender, dtype: int64
```

In [29]:
```
NewCustomerList[NewCustomerList.gender == "U"]
```

Out[29]:

	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job_title	job_industry_category	wealth_segment	deceased_indicator	owns
59	Normy	Goodinge	U	5	NaT	Associate Professor	IT	Mass Customer	N	
226	Hatti	Carletti	U	35	NaT	Legal Assistant	IT	Affluent Customer	N	
324	Rozamond	Turtle	U	69	NaT	Legal Assistant	IT	Mass Customer	N	
358	Tamas	Swatman	U	65	NaT	Assistant Media Planner	Entertainment	Affluent Customer	N	
360	Tracy	Andrejevic	U	71	NaT	Programmer II	IT	Mass Customer	N	
374	Agneta	McAmish	U	66	NaT	Structural Analysis Engineer	IT	Mass Customer	N	
434	Gregg	Aimeric	U	52	NaT	Internal Auditor	IT	Mass Customer	N	
439	Johna	Bunker	U	93	NaT	Tax Accountant	IT	Mass Customer	N	
574	Harlene	Nono	U	69	NaT	Human Resources Manager	IT	Mass Customer	N	
598	Gerianne	Kaysor	U	15	NaT	Project Manager	IT	Affluent Customer	N	
664	Chicky	Sinclar	U	43	NaT	Operator	IT	High Net Worth	N	
751	Adriana	Saundercock	U	20	NaT	Nurse	IT	High Net Worth	N	
775	Dmitri	Viant	U	62	NaT	Paralegal	Financial Services	Affluent Customer	N	
835	Porty	Hansed	U	88	NaT	General Manager	IT	Mass Customer	N	
883	Shara	Bramhill	U	24	NaT	NaN	IT	Affluent Customer	N	
904	Roth	Crum	U	0	NaT	Legal Assistant	IT	Mass Customer	N	
984	Pauline	Dallosso	U	82	NaT	Desktop Support Technician	IT	Affluent Customer	N	

In [30]:
```
NewCustomerList['DOB'].value_counts()
```

Out[30]:
```
1961-07-31    2
1994-04-15    2
1987-01-15    2
1951-11-28    2
1979-07-28    2
             ..
1947-04-21    1
1995-10-10    1
1964-08-23    1
1962-05-12    1
1958-04-21    1
Name: DOB, Length: 958, dtype: int64
```

In [31]:
```
NewCustomerList['job_industry_category'].value_counts()
```

Out[31]:
```
Financial Services    203
Manufacturing         199
Health                152
Retail                 78
Property               64
IT                     51
Entertainment          37
Argiculture            26
Telecommunications     25
Name: job_industry_category, dtype: int64
```

In [32]:
```
NewCustomerList['job_industry_category'].value_counts()
```

Out[32]:
```
Financial Services    203
Manufacturing         199
Health                152
Retail                 78
Property               64
IT                     51
Entertainment          37
Argiculture            26
Telecommunications     25
Name: job_industry_category, dtype: int64
```

In [33]:
```
NewCustomerList['wealth_segment'].value_counts()
```

Out[33]:
```
Mass Customer        508
High Net Worth       251
Affluent Customer    241
Name: wealth_segment, dtype: int64
```

In [34]:
```
NewCustomerList['state'].value_counts()
```

Out[34]:
```
NSW    506
VIC    266
QLD    228
Name: state, dtype: int64
```

In [35]:
```
NewCustomerList['owns_car'].value_counts()
```

Out[35]:
```
No     507
Yes    493
Name: owns_car, dtype: int64
```

In [36]:
```
NewCustomerList['deceased_indicator'].value_counts()
```

Out[36]:
```
N    1000
Name: deceased_indicator, dtype: int64
```

3. Customer Demographic

In [37]:
```
CustomerDemographic.head()
```

Out[37]:

	customer_id	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job_title	job_industry_category	wealth_segment	deceased_ind
0	1	Laraine	Medendorp	F	93	1953-10-12	Executive Secretary	Health	Mass Customer	
1	2	Eli	Bockman	Male	81	1980-12-16	Administrative Officer	Financial Services	Mass Customer	
2	3	Arlin	Dearle	Male	61	1954-01-20	Recruiting Manager	Property	Mass Customer	
3	4	Talbot	NaN	Male	33	1961-10-03	NaN	IT	Mass Customer	
4	5	Sheila-kathryn	Calton	Female	56	1977-05-13	Senior Editor	NaN	Affluent Customer	

In [38]:
```
CustomerDemographic.dtypes
```

Out[38]:
```
customer_id                             int64
first_name                             object
last_name                              object
gender                                 object
past_3_years_bike_related_purchases     int64
DOB                            datetime64[ns]
job_title                              object
job_industry_category                  object
wealth_segment                         object
deceased_indicator                     object
default                                object
owns_car                               object
tenure                                float64
dtype: object
```

In [39]:
```
#Checking for null values
CustomerDemographic.isnull().sum()
```

Out[39]:
```
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
```

In [40]:
```
#Checking for duplicate data
CustomerDemographic.duplicated().sum()
```

Out[40]:
```
0
```

In [41]:
```
#Checking for uniqueness of each column
CustomerDemographic.nunique()
```

Out[41]:
```
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
```

In [42]:
```
# Exploring Column
CustomerDemographic.columns
```

Out[42]:
```
Index(['customer_id', 'first_name', 'last_name', 'gender',
       'past_3_years_bike_related_purchases', 'DOB', 'job_title',
       'job_industry_category', 'wealth_segment', 'deceased_indicator',
       'default', 'owns_car', 'tenure'],
      dtype='object')
```

In [43]:
```
CustomerDemographic['gender'].value_counts()
```

Out[43]:
```
Female    2037
Male      1872
U           88
M            1
Femal        1
F            1
Name: gender, dtype: int64
```

In [44]:
```
#Re-naming the categories
CustomerDemographic['gender'] = CustomerDemographic['gender'].replace('F','Female').replace('M','Male').replace('Femal','Female')
```

In [45]:
```
CustomerDemographic['gender'].value_counts()
```

Out[45]:
```
Female    2039
Male      1873
U           88
Name: gender, dtype: int64
```

In [46]:
```
CustomerDemographic['past_3_years_bike_related_purchases'].value_counts()
```

Out[46]:
```
16    56
19    56
20    54
67    54
2     50
      ..
8     28
86    27
95    27
85    27
92    24
Name: past_3_years_bike_related_purchases, Length: 100, dtype: int64
```

In [47]:
```
CustomerDemographic['DOB'].value_counts()
```

Out[47]:
```
1978-01-30    7
1977-05-13    4
1976-07-16    4
1978-08-19    4
1976-09-25    4
              ..
1974-06-16    1
1969-11-21    1
1961-11-24    1
1995-05-03    1
1972-04-14    1
Name: DOB, Length: 3448, dtype: int64
```

In [48]:
```
CustomerDemographic['job_title'].value_counts()
```

Out[48]:
```
Business Systems Development Analyst    45
Social Worker                           44
Tax Accountant                          44
Internal Auditor                        42
Legal Assistant                         41
                                        ..
Geologist II                             4
Research Assistant III                   3
Health Coach III                         3
Health Coach I                           3
Developer I                              1
Name: job_title, Length: 195, dtype: int64
```

In [49]:
```
CustomerDemographic['job_industry_category'].value_counts()
```

Out[49]:
```
Manufacturing         799
Financial Services    774
Health                602
Retail                358
Property              267
IT                    223
Entertainment         136
Argiculture           113
Telecommunications     72
Name: job_industry_category, dtype: int64
```

In [50]:
```
CustomerDemographic['wealth_segment'].value_counts()
```

Out[50]:
```
Mass Customer        2000
High Net Worth       1021
Affluent Customer     979
Name: wealth_segment, dtype: int64
```

In [51]:
```
CustomerDemographic['deceased_indicator'].value_counts()
```

Out[51]:
```
N    3998
Y       2
Name: deceased_indicator, dtype: int64
```

In [52]:
```
CustomerDemographic['default'].value_counts()
```

Out[52]:
```
100                                          113
1                                            112
-1                                           111
-100                                          99
Ù¡Ù¢Ù£                                        53
                                            ...
<img src=x onerror=alert('hi') />             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
```

In [53]:
```
# The values are inconsistent, hence dropping the column.
CustomerDemographic = CustomerDemographic.drop('default', axis=1)
```

In [54]:
```
CustomerDemographic.head()
```

Out[54]:

	customer_id	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job_title	job_industry_category	wealth_segment	deceased_ind
0	1	Laraine	Medendorp	Female	93	1953-10-12	Executive Secretary	Health	Mass Customer	
1	2	Eli	Bockman	Male	81	1980-12-16	Administrative Officer	Financial Services	Mass Customer	
2	3	Arlin	Dearle	Male	61	1954-01-20	Recruiting Manager	Property	Mass Customer	
3	4	Talbot	NaN	Male	33	1961-10-03	NaN	IT	Mass Customer	
4	5	Sheila-kathryn	Calton	Female	56	1977-05-13	Senior Editor	NaN	Affluent Customer	

In [55]:
```
CustomerDemographic['owns_car'].value_counts()
```

Out[55]:
```
Yes    2024
No     1976
Name: owns_car, dtype: int64
```

In [56]:
```
CustomerDemographic['tenure'].value_counts()
```

Out[56]:
```
7.0     235
5.0     228
11.0    221
10.0    218
16.0    215
8.0     211
18.0    208
12.0    202
9.0     200
14.0    200
6.0     192
13.0    191
4.0     191
17.0    182
15.0    179
1.0     166
3.0     160
19.0    159
2.0     150
20.0     96
22.0     55
21.0     54
Name: tenure, dtype: int64
```

4. Customer Address

In [57]:
```
CustomerAddress.head()
```

Out[57]:

	customer_id	address	postcode	state	country	property_valuation
0	1	060 Morning Avenue	2016	New South Wales	Australia	10
1	2	6 Meadow Vale Court	2153	New South Wales	Australia	10
2	4	0 Holy Cross Court	4211	QLD	Australia	9
3	5	17979 Del Mar Point	2448	New South Wales	Australia	4
4	6	9 Oakridge Court	3216	VIC	Australia	9

In [58]:
```
CustomerAddress.dtypes
```

Out[58]:
```
customer_id           int64
address              object
postcode              int64
state                object
country              object
property_valuation    int64
dtype: object
```

In [59]:
```
CustomerAddress.shape
```

Out[59]:
```
(3999, 6)
```

In [60]:
```
#Checking for null values.
CustomerAddress.isnull().sum()
```

Out[60]:
```
customer_id          0
address              0
postcode             0
state                0
country              0
property_valuation   0
dtype: int64
```

In [61]:
```
#Checking for duplicate values
CustomerAddress.duplicated().sum()
```

Out[61]:
```
0
```

In [62]:
```
#Checking for uniqueness of each column
CustomerAddress.nunique()
```

Out[62]:
```
customer_id          3999
address              3996
postcode              873
state                   5
country                 1
property_valuation     12
dtype: int64
```

In [63]:
```
# Exploring Column
CustomerAddress['postcode'].value_counts()
```

Out[63]:
```
2170    31
2155    30
2145    30
2153    29
2770    26
        ..
4552     1
4555     1
2485     1
3580     1
4421     1
Name: postcode, Length: 873, dtype: int64
```

In [64]:
```
CustomerAddress['state'].value_counts()
```

Out[64]:
```
NSW                2054
VIC                 939
QLD                 838
New South Wales      86
Victoria             82
Name: state, dtype: int64
```

In [65]:
```
CustomerAddress['country'].value_counts()
```

Out[65]:
```
Australia    3999
Name: country, dtype: int64
```

In [66]:
```
CustomerAddress['property_valuation'].value_counts()
```

Out[66]:
```
9     647
8     646
10    577
7     493
11    281
6     238
5     225
4     214
12    195
3     186
1     154
2     143
Name: property_valuation, dtype: int64
```

All the columns appear to have consistent and correct information