In [1]:
```python
import warnings
warnings.filterwarnings('ignore')
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```python
df=pd.ExcelFile('KPMG_final.xlsx')
df1=pd.read_excel(df,'Transactions')
df2=pd.read_excel(df,'NewCustomerList')
df3=pd.read_excel(df,'CustomerDemographic')
df4=pd.read_excel(df,'CustomerAddress')
```

Data Quality issues in df1 - Transactions

In [3]:
```python
df1.to_csv('Transactions.csv')
```

In [4]:
```python
df1.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_cost	product_first_sold_date
0	1	2	2950	2017-02-25	0.0	Approved	Solex	Standard	medium	medium	71.49	53.62	41245.0
1	2	3	3120	2017-05-21	1.0	Approved	Trek Bicycles	Standard	medium	large	2091.47	388.92	41701.0
2	3	37	402	2017-10-16	0.0	Approved	OHM Cycles	Standard	low	medium	1793.43	248.82	36361.0
3	4	88	3135	2017-08-31	0.0	Approved	Norco Bicycles	Standard	medium	medium	1198.46	381.10	36145.0
4	5	78	787	2017-10-01	1.0	Approved	Giant Bicycles	Standard	medium	large	1765.30	709.48	42226.0

In [5]:
```python
df1.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
```

In [6]:
```python
#shape of the data
df1.shape
```

Out[6]:
```
(20000, 13)
```

In [7]:
```python
#check for null values
df1.isnull().sum()
```

Out[7]:
```
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
```

online_order,brand,product_line,product_class,product_size,standard_cost,product_first_sold_date are the columns with missing values.Lets do some analysis in those columns.

In [8]:
```python
#check if there are any duplicate values
df1.duplicated().sum()
```

Out[8]:
```
0
```

There are no duplicate values in the transactions dataframe which means that the dataset is unique.

In [9]:
```python
#uniqueness of the data
df1.nunique()
```

Out[9]:
```
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 [10]:
```python
#exploring and understanding the columns
df1.columns
```

Out[10]:
```
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 [11]:
```python
#number of approved and cancelled orders
df1['order_status'].value_counts()
```

Out[11]:
```
Approved     19821
Cancelled      179
Name: order_status, dtype: int64
```

In [12]:
```python
#different brands
df1['brand'].value_counts()
```

Out[12]:
```
Solex             4253
Giant Bicycles    3312
WeareA2B          3295
OHM Cycles        3043
Trek Bicycles     2990
Norco Bicycles    2910
Name: brand, dtype: int64
```

In [13]:
```python
#different product lines
df1['product_line'].value_counts()
```

Out[13]:
```
Standard    14176
Road         3970
Touring      1234
Mountain      423
Name: product_line, dtype: int64
```

In [14]:
```python
#different product classes
df1['product_class'].value_counts()
```

Out[14]:
```
medium    13826
high       3013
low        2964
Name: product_class, dtype: int64
```

In [15]:
```python
#different product sizes
df1['product_size'].value_counts()
```

Out[15]:
```
medium    12990
large      3976
small      2837
Name: product_size, dtype: int64
```

In [16]:
```python
#accuracy check
df1['product_first_sold_date'].value_counts()
```

Out[16]:
```
33879.0    234
41064.0    229
37823.0    227
39880.0    222
38216.0    220
          ...
41848.0    169
42404.0    168
41922.0    166
37659.0    163
34586.0    162
Name: product_first_sold_date, Length: 100, dtype: int64
```

In [17]:
```python
#product_first_sold_date is in the form of numbers.Let's convert it into date.
df1['product_first_sold_date'] = pd.to_datetime(df1['product_first_sold_date'],unit='s')
```

In [18]:
```python
df1['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]:
```python
df1['product_first_sold_date'].head(25)
```

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
10   1970-01-01 11:19:44
11   1970-01-01 11:42:52
12   1970-01-01 09:35:27
13   1970-01-01 09:36:26
14   1970-01-01 10:36:33
15   1970-01-01 10:31:13
16   1970-01-01 10:36:46
17   1970-01-01 09:24:48
18   1970-01-01 11:05:15
19   1970-01-01 10:22:17
20   1970-01-01 10:05:34
21   1970-01-01 10:06:01
22   1970-01-01 11:42:25
23   1970-01-01 11:46:44
24   1970-01-01 09:27:59
Name: product_first_sold_date, dtype: datetime64[ns]
```

The column 'product_first_sold_date' seems to be wrong since it shows the same date with different timestamps.This looks inaccurate.

Data Quality issues in df2- NewCustomerList

In [20]:
```python
df2.to_csv('NewCustomerList.csv')
```

In [21]:
```python
df2.head()
```

Out[21]:

	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job_title	job_industry_category	wealth_segment	deceased_indicator	owns_car	...	state	country	property_valuation	Unname
0	Chickie	Brister	Male	86	1957-07-12	General Manager	Manufacturing	Mass Customer	N	Yes	...	QLD	Australia	6	0.
1	Morly	Genery	Male	69	1970-03-22	Structural Engineer	Property	Mass Customer	N	No	...	NSW	Australia	11	0.
2	Ardelis	Forrester	Female	10	1974-08-28	Senior Cost Accountant	Financial Services	Affluent Customer	N	No	...	VIC	Australia	5	0.
3	Lucine	Stutt	Female	64	1979-01-28	Account Representative III	Manufacturing	Affluent Customer	N	Yes	...	QLD	Australia	1	0.
4	Melinda	Hadlee	Female	34	1965-09-21	Financial Analyst	Financial Services	Affluent Customer	N	No	...	NSW	Australia	9	0.

5 rows × 23 columns

In [22]:
```python
df2.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
```

In [23]:
```python
#lets remove the unwanted columns in the dataset
df2.drop(['Unnamed: 16','Unnamed: 17','Unnamed: 18','Unnamed: 19','Unnamed: 20'],axis=1,inplace=True)
```

In [24]:
```python
df2.shape
```

Out[24]:
```
(1000, 18)
```

In [25]:
```python
df2.isnull().sum()
```

Out[25]:
```
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
```

last_name,DOB,job_title,job_industry_category are the columns that has missing values. Lets perform analysis on these columns of the dataset.

In [26]:
```python
df2.duplicated().sum()
```

Out[26]:
```
0
```

The NewCustomerList doesnt have any duplicated values which means that the dataset is unique.

In [27]:
```python
df2.nunique()
```

Out[27]:
```
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
```

In [28]:
```python
#columns of NewCustomerList
df2.columns
```

Out[28]:
```
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 [29]:
```python
#check the column gender since it says that the unique values are 3.
df2['gender'].value_counts()
```

Out[29]:
```
Female    513
Male      470
U          17
Name: gender, dtype: int64
```

In [30]:
```python
#Lets remove the values that has 'U' in it since it doesnt apply to any gender and is irrelevant.
df2 = df2[df2["gender"].str.contains("U") == False]
```

In [31]:
```python
#check if the column values of U are removed
df2['gender'].value_counts()
```

Out[31]:
```
Female    513
Male      470
Name: gender, dtype: int64
```

In [32]:
```python
#different job industry categories
df2['job_industry_category'].value_counts()
```

Out[32]:
```
Financial Services    202
Manufacturing         199
Health                152
Retail                 78
Property               64
IT                     36
Entertainment          36
Argiculture            26
Telecommunications     25
Name: job_industry_category, dtype: int64
```

In [33]:
```python
#different wealth segments
df2['wealth_segment'].value_counts()
```

Out[33]:
```
Mass Customer        499
High Net Worth       249
Affluent Customer    235
Name: wealth_segment, dtype: int64
```

In [34]:
```python
#different states
df2['state'].value_counts()
```

Out[34]:
```
NSW    499
VIC    258
QLD    226
Name: state, dtype: int64
```

Data Quality issues in df3-CustomerDemographic

In [35]:
```python
df3.to_csv('CustomerDemographic.csv')
```

In [36]:
```python
df3.head()
```

Out[36]:

	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
0	1	Laraine	Medendorp	F	93	1953-10-12	Executive Secretary	Health	Mass Customer	N	'"'	Yes	11.0
1	2	Eli	Bockman	Male	81	1980-12-16	Administrative Officer	Financial Services	Mass Customer	N	<script>alert('hi')</script>	Yes	16.0
2	3	Arlin	Dearle	Male	61	1954-01-20	Recruiting Manager	Property	Mass Customer	N	2018-02-01 00:00:00	Yes	15.0
3	4	Talbot	NaN	Male	33	1961-10-03	NaN	IT	Mass Customer	N	() { _; } >_[(())] { touch /tmp/blns.shellsh...	No	7.0
4	5	Sheila-kathryn	Calton	Female	56	1977-05-13	Senior Editor	NaN	Affluent Customer	N	NIL	Yes	8.0

In [37]:
```python
df3.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
```

In [38]:
```python
df3.shape
```

Out[38]:
```
(4000, 13)
```

In [39]:
```python
df3.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
```

last_name,DOB,job_title,job_industry_category,default and tenure are the columns with null values. lets Perform some analysis on these columns.

In [40]:
```python
df3.duplicated().sum()
```

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

df3 doesnt have any duplicate value so its a unique dataset.

In [41]:
```python
df3.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]:
```python
#columns of CustomerDemographic
df3.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]:
```python
#df3 has 6 different genders mentioned.Lets analyse and drop the unwanted values
df3.gender.value_counts()
```

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

In [44]:
```python
#U is not a gender.Lets drop the rows with U values.
df3 = df3[df3["gender"].str.contains("U") == False]
```

In [45]:
```python
#F and Femal refers to female so lets tag them into Female and M to Male
df3['gender']=df3['gender'].replace('F','Female').replace('Femal','Female').replace('M','Male')
```

In [46]:
```python
df3.gender.value_counts()
```

Out[46]:
```
Female    2039
Male      1873
Name: gender, dtype: int64
```

In [47]:
```python
df3.wealth_segment.value_counts()
```

Out[47]:
```
Mass Customer        1954
High Net Worth        996
Affluent Customer     962
Name: wealth_segment, dtype: int64
```

In [48]:
```python
df3.job_industry_category.value_counts()
```

Out[48]:
```
Manufacturing         796
Financial Services    767
Health                596
Retail                358
Property              267
IT                    151
Entertainment         136
Argiculture           113
Telecommunications     72
Name: job_industry_category, dtype: int64
```

In [49]:
```python
df3.default.value_counts()
```

Out[49]:
```
100                                        113
1                                          112
-1                                         111
-100                                        99
Ù¡Ù¢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
```

In [50]:
```python
#The column default has irrelevant values and will not affect the dataset if deleted so lets remove the column.
df3.drop(['default'],axis=1)
```

Out[50]:

	customer_id	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job_title	job_industry_category	wealth_segment	deceased_indicator	owns_car	tenure
0	1	Laraine	Medendorp	Female	93	1953-10-12	Executive Secretary	Health	Mass Customer	N	Yes	11.0
1	2	Eli	Bockman	Male	81	1980-12-16	Administrative Officer	Financial Services	Mass Customer	N	Yes	16.0
2	3	Arlin	Dearle	Male	61	1954-01-20	Recruiting Manager	Property	Mass Customer	N	Yes	15.0
3	4	Talbot	NaN	Male	33	1961-10-03	NaN	IT	Mass Customer	N	No	7.0
4	5	Sheila-kathryn	Calton	Female	56	1977-05-13	Senior Editor	NaN	Affluent Customer	N	Yes	8.0
...	...	...	...	...	...	...	...	...	...	...	...	...
3994	3995	Rusty	Iapico	Male	93	1975-12-12	Staff Scientist	Manufacturing	Mass Customer	N	Yes	14.0
3995	3996	Rosalia	Halgarth	Female	8	1975-08-09	VP Product Management	Health	Mass Customer	N	No	19.0
3996	3997	Blanch	Nisuis	Female	87	2001-07-13	Statistician II	Manufacturing	High Net Worth	N	Yes	1.0
3998	3999	Patrizius	NaN	Male	11	1973-10-24	NaN	Manufacturing	Affluent Customer	N	Yes	10.0
3999	4000	Kippy	Oldland	Male	76	1991-11-05	Software Engineer IV	NaN	Affluent Customer	N	No	11.0

3912 rows × 12 columns

In [51]:
```python
#deceased_indicato refers to the people who are alive and not alive.
df3.deceased_indicator.value_counts()
```

Out[51]:
```
N    3910
Y       2
Name: deceased_indicator, dtype: int64
```

In [52]:
```python
#We need the data of only the people who are alive so lets remove the row values with 'Y'
df3 = df3[df3["deceased_indicator"].str.contains("Y") == False]
```

In [53]:
```python
df3.deceased_indicator.value_counts()
```

Out[53]:
```
N    3910
Name: deceased_indicator, dtype: int64
```

Data quality issues in df4-CustomerAddress

In [54]:
```python
df4.to_csv('CustomerAddress.csv')
```

In [55]:
```python
df4.head()
```

Out[55]:

	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 [56]:
```python
df4.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
```

In [57]:
```python
df4.shape
```

Out[57]:
```
(3999, 6)
```

In [58]:
```python
df4.isnull().sum()
```

Out[58]:
```
customer_id           0
address               0
postcode              0
state                 0
country               0
property_valuation    0
dtype: int64
```

In [59]:
```python
df4.nunique()
```

Out[59]:
```
customer_id           3999
address               3996
postcode               873
state                    5
country                  1
property_valuation      12
dtype: int64
```

In [60]:
```python
df4.duplicated().sum()
```

Out[60]:
```
0
```

No duplicated values so its unique

In [61]:
```python
df4.state.value_counts()
```

Out[61]:
```
NSW                2054
VIC                 939
QLD                 838
New South Wales      86
Victoria             82
Name: state, dtype: int64
```

In [62]:
```python
#there are 5 columns- NSW,VIC,QLD,New South Wales and Victoria. Lets change New south Wales to NSW and Victoria to VIC.
df4['state']=df4['state'].replace('New South Wales','NSW').replace('Victoria','VIC')
```

In [63]:
```python
df4.state.value_counts()
```

Out[63]:
```
NSW    2140
VIC    1021
QLD     838
Name: state, dtype: int64
```

In [65]:
```python
df4.property_valuation.value_counts()
```

Out[65]:
```
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
```