In [146…

```
# importing library to conduct data analysis
import pandas as pd

# reading the given KPMG data set excel
df = pd.ExcelFile("Downloads/KPMG data file.xlsx")
df1 = pd.read_excel(df, "Transactions") # reading Transactions sheet
df2 = pd.read_excel(df, "NewCustomerList") # reading NewCustomerList
df3 = pd.read_excel(df, "CustomerDemographic") # reading CustomerDemographic
df4 = pd.read_excel(df, "CustomerAddress") # reading CustomerAddress sheet
```

```
/var/folders/yd/2fxr4bx52nzf_pdbhh3cqqwh0000gn/T/ipykernel_781/3218012014.p
y:7: FutureWarning: Inferring datetime64[ns] from data containing strings is
deprecated and will be removed in a future version. To retain the old behavi
or explicitly pass Series(data, dtype=datetime64[ns])
  df2 = pd.read_excel(df, "NewCustomerList") # reading NewCustomerList
/var/folders/yd/2fxr4bx52nzf_pdbhh3cqqwh0000gn/T/ipykernel_781/3218012014.p
y:8: FutureWarning: Inferring datetime64[ns] from data containing strings is
deprecated and will be removed in a future version. To retain the old behavi
or explicitly pass Series(data, dtype=datetime64[ns])
  df3 = pd.read_excel(df, "CustomerDemographic") # reading CustomerDemograph
ic sheet
```

In [147…

```
# Reviewing Transactions dataset and checking problems
df1.head(10)
```

Out[147]:

	transaction_id	product_id	customer_id	transaction_date	online_order	order_status	
0	1	2	2950	2017-02-25	0.0	Approved	
1	2	3	3120	2017-05-21	1.0	Approved	
2	3	37	402	2017-10-16	0.0	Approved	
3	4	88	3135	2017-08-31	0.0	Approved	
4	5	78	787	2017-10-01	1.0	Approved	
5	6	25	2339	2017-03-08	1.0	Approved	
6	7	22	1542	2017-04-21	1.0	Approved	V
7	8	15	2459	2017-07-15	0.0	Approved	V
8	9	67	1305	2017-08-10	0.0	Approved	
9	10	12	3262	2017-08-30	1.0	Approved	V

In [148…

```
df1.info() # generating an overview of the df1's (Transactions) structure ar
```

```
<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 [149…

```
# checking the number of rows and columns of df1
df1.shape
```

Out[149]:

```
(20000, 13)
```

In [150…

```
# checking that whether there are any null values in df1
df1.isnull().sum()
```

Out[150]:

```
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 360 values missing in column 4, and there are 197 values missing in column 6,7,8,9,11,12 respectively.

In [151…

```
# checking for duplication in df1
df1.duplicated().sum()
```

Out[151]:

```
0
```

This is fine. There are no duplicated rows in Transactions sheet.

In [152…

```
# Checking for uniqueness of each column in df1
df1.nunique()
```

Out[152]:

```
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 [153…

```
# Reviewing the columns of df1
df1.columns
```

Out[153]:

```
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 [154…

```
# checking the values of "order_status"
df1["order_status"].value_counts()
```

Out[154]:

```
Approved     19821
Cancelled      179
Name: order_status, dtype: int64
```

In [155…

```
# checking the values of "brand"
df1["brand"].value_counts()
```

Out[155]:

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

In [156…

```
# checking the values of "product_line"
df1["product_line"].value_counts()
```

Out[156]:

```
Standard    14176
Road         3970
Touring      1234
Mountain      423
Name: product_line, dtype: int64
```

In [157…

```
# checking the values of "product_class"
df1["product_class"].value_counts()
```

Out[157]:

```
medium    13826
high       3013
low        2964
Name: product_class, dtype: int64
```

In [158…

```
# checking the values of "product_size"
df1["product_size"].value_counts()
```

Out[158]:

```
medium    12990
large      3976
small      2837
Name: product_size, dtype: int64
```

In [159…

```
# checking the values of "product_first_sold_date"
df1["product_first_sold_date"].value_counts()
```

Out[159]:

```
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 [160…

```
# converting the intergers of product_first_sold_date column to "datetime" c
df1["product_first_sold_date"] = pd.to_datetime(df1["product_first_sold_date
df1["product_first_sold_date"].head(10)
```

Out[160]:

```
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 product_first_sold_date are all integers. Need to be converted to "datetime" object.

In [161…

```
df1["product_first_sold_date"].head(30)
```

Out[161]:

```
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
25    1970-01-01 11:42:25
26    1970-01-01 11:24:07
27    1970-01-01 11:49:20
28    1970-01-01 11:51:50
29    1970-01-01 11:38:42
Name: product_first_sold_date, dtype: datetime64[ns]
```

There're errors in the column of product_first_sold_date, as the values in this column show that the product's first sold dates are on the same day, just on different times of the day.

In [162…

```
# Reviewing NewCustomerList dataset and checking problems
df2.head(10)
```

Out[162]:

	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job_titl
0	Marinna	Kauschke	Female	21	1973-03-15	Sale Associat
1	Olia	O' Mullan	Female	77	1973-03-24	Accour Executiv
2	Brigitte	Whellams	Female	67	1973-05-09	Paymer Adjustmer Coordinatc
3	Ivy	Farr	Female	56	1973-07-03	Offic Assistant I'
4	Beverlee	Ungerechts	Female	49	1973-10-03	Civ Enginee
5	Skipp	Swales	Male	15	1973-11-14	Communit Outreac Specialis
6	Leighton	Firbanks	Male	51	1973-12-22	Teache
7	Claudetta	Ricciardiello	Female	61	1974-04-30	Interna Auditc
8	Harland	Messenger	Male	90	1974-05-28	Softwar Tes Engineer
9	Babara	Sissel	Female	50	1974-06-08	Nal

10 rows × 23 columns

In [163…

```
df2.info() # generating an overview of the df2's (NewCustomerList) structure
```

```
<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 [164…

```
# dropping the Unnamed columns from df2
df2.drop(["Unnamed: 16","Unnamed: 17","Unnamed: 18","Unnamed: 19","Unnamed:
```

Need to drop the columns with unexpected errors. There are columns with names of "Unnamed".

In [165…

```
# checking the number of rows and columns of df2
df2.shape
```

Out[165]:

```
(1000, 18)
```

In [166…

```
# checking that whether there are any null values in df2
df2.isnull().sum()
```

Out[166]:

```
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
```

There are 29 missing values in column of last_name, 17 missing values in column of DOB, 106 missing values in column of job_title, and 165 missing values in column of job_industry_category.

In [167…

```
# checking for duplication in df2
df2.duplicated().sum()
```

Out[167]:

```
0
```

This is fine. There are no duplicated rows in NewCustomerList sheet.

In [168…

```
# Checking for uniqueness of each column in df2
df2.nunique()
```

Out[168]:

```
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 [169…

```
# Reviewing the columns of df2
df2.columns
```

Out[169]:

```
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 [170…

```
# checking the values of "gender"
df2["gender"].value_counts()
```

Out[170]:

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

In [171…

```
# checking the U values of "gender"
df2[df2["gender"] == "U"]
```

Out[171]:

	first_name	last_name	gender	past_3_years_bike_related_purchases	DOB	job_
983	Normy	Goodinge	U	5	NaT	Assoc Profe
984	Hatti	Carletti	U	35	NaT	L Assis
985	Rozamond	Turtle	U	69	NaT	L Assis
986	Tamas	Swatman	U	65	NaT	Assis M Pla
987	Tracy	Andrejevic	U	71	NaT	Program
988	Agneta	McAmish	U	66	NaT	Struc Ana Engi
989	Gregg	Aimeric	U	52	NaT	Inte Au
990	Johna	Bunker	U	93	NaT	Accour
991	Harlene	Nono	U	69	NaT	Hu Resou Man
992	Gerianne	Kaysor	U	15	NaT	Pr Man
993	Chicky	Sinclar	U	43	NaT	Oper
994	Adriana	Saundercock	U	20	NaT	N
995	Dmitri	Viant	U	62	NaT	Paral
996	Porty	Hansed	U	88	NaT	Ger Man
997	Shara	Bramhill	U	24	NaT	
998	Roth	Crum	U	0	NaT	L Assis
999	Pauline	Dallosso	U	82	NaT	Des Sup Techn

The above 17 rows are information about customers with unknow gender.

In [172…

```
# checking the values of "DOB"
df2["DOB"].value_counts()
```

Out[172]:

```
1965-07-03    2
1974-12-25    2
1941-07-21    2
1977-11-08    2
1978-12-14    2
             ..
1959-12-25    1
1960-01-21    1
1960-02-14    1
1960-03-18    1
2002-02-27    1
Name: DOB, Length: 958, dtype: int64
```

In [173…

```
# checking the values of "job_industry_category"
df2["job_industry_category"].value_counts()
```

Out[173]:

```
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 [174…

```
# checking the values of "wealth_segment"
df2["wealth_segment"].value_counts()
```

Out[174]:

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

In [175…

```
# checking the values of "deceased_indicator"
df2["deceased_indicator"].value_counts()
```

Out[175]:

```
N    1000
Name: deceased_indicator, dtype: int64
```

In [176…

```
# checking the values of "owns_car"
df2["owns_car"].value_counts()
```

Out[176]:

```
No     507
Yes    493
Name: owns_car, dtype: int64
```

In [177…

```
# checking the values of "state"
df2["state"].value_counts()
```

Out[177]:

```
NSW    506
VIC    266
QLD    228
Name: state, dtype: int64
```

In [178…

```
# Reviewing CustomerDemographic dataset and checking problems
df3.head(10)
```

Out[178]:

	customer_id	first_name	last_name	gender	past_3_years_bike_related_purchases	D
0	34	Jephthah	Bachmann	U	59	184 12-
1	720	Darrel	Canet	Male	67	193 10-
2	1092	Katlin	Creddon	Female	56	193 0
3	3410	Merrili	Brittin	Female	93	194 0
4	2413	Abbey	Murrow	Male	27	194 08-
5	658	Donn	Bonnell	Male	38	194 01-
6	1243	Robbert	Blakey	Male	73	195 0
7	1565	Jay	Janiszewski	Male	71	195 08-
8	1177	Bobbette	Pozzi	Female	47	195 08-
9	3471	Brita	Afonso	Female	95	195 0

In [179…

```
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 [180…

```
df3.isnull().sum()
```

Out[180]:

```
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
```

There are 125 missing values in column of last_name, 87 missing values in column of DOB, 506 missing values in column of job_title, 656 missing values in column of job_industry_category, 302 missing values in column of default, and 87 missing values in column of tenure.

In [181…

```
df3.duplicated().sum()
```

Out[181]:

```
0
```

This is fine. There are no duplicated rows in sheet of CustomerDemographic.

In [182…

```
df3.nunique()
```

Out[182]:

```
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 [183…

```
df3.columns
```

Out[183]:

```
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 [184…

```
df3["gender"].value_counts()
```

Out[184]:

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

Some of the values in the column of gender are not properly recorded. Need to rename "F" and "Femal" with "Female", and "M" with "Male".

In [185…

```
df3["gender"] = df3["gender"].replace("F", "Female").replace("Femal", "Femal
df3["gender"]
```

Out[185]:

```
0            U
1         Male
2       Female
3       Female
4         Male
         ...  
3995         U
3996         U
3997         U
3998         U
3999         U
Name: gender, Length: 4000, dtype: object
```

In [186…

```
df3["gender"].value_counts()
```

Out[186]:

```
Female    2039
Male      1873
U           88
Name: gender, dtype: int64
```

In [187…

```
df3[df3["gender"] == "U"]
```

Out[187]:

	customer_id	first_name	last_name	gender	past_3_years_bike_related_purchases
0	34	Jephthah	Bachmann	U	59
3913	144	Jory	Barrabeale	U	71
3914	168	Reggie	Broggetti	U	8
3915	267	Edgar	Buckler	U	53
3916	290	Giorgio	Kevane	U	42
...	...	...	...	...	...
3995	3779	Ulick	Daspar	U	68
3996	3883	Nissa	Conrad	U	35
3997	3931	Kylie	Epine	U	19
3998	3935	Teodor	Alfonsini	U	72
3999	3998	Sarene	Woolley	U	60

88 rows × 13 columns

The above 88 rows are information about customers with unknow gender.

In [188…

```
df3["past_3_years_bike_related_purchases"].value_counts()
```

Out[188]:

```
19    56
16    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 [189…

```
df3["DOB"].value_counts()
```

Out[189]:

```
1978-01-30    7
1976-07-16    4
1978-08-19    4
1976-09-25    4
1964-07-08    4
             ..
1972-06-05    1
1972-06-21    1
1972-07-11    1
1972-07-17    1
2002-03-11    1
Name: DOB, Length: 3448, dtype: int64
```

In [190…

```
df3["job_title"].value_counts()
```

Out[190]:

```
Business Systems Development Analyst    45
Social Worker                           44
Tax Accountant                          44
Internal Auditor                        42
Recruiting Manager                      41
                                        ..
Human Resources Assistant IV             4
Research Assistant III                   3
Health Coach I                           3
Health Coach III                         3
Developer I                              1
Name: job_title, Length: 195, dtype: int64
```

In [191…

```
df3["job_industry_category"].value_counts()
```

Out[191]:

```
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 [192…

```
df3["wealth_segment"].value_counts()
```

Out[192]:

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

In [193…

```
df3["deceased_indicator"].value_counts()
```

Out[193]:

```
N    3998
Y       2
Name: deceased_indicator, dtype: int64
```

In [194…

```
df3["default"].value_counts()
```

Out[194]:

```
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
```

We note that the column of default has inconsistent values, so we drop this column.

In [195…

```
df3.drop(["default"], axis = 1, inplace = True)
df3.head(10)
```

Out[195]:

	customer_id	first_name	last_name	gender	past_3_years_bike_related_purchases	DO
0	34	Jephthah	Bachmann	U	59	184 12-
1	720	Darrel	Canet	Male	67	193 10-
2	1092	Katlin	Creddon	Female	56	193 C
3	3410	Merrili	Brittin	Female	93	194 C
4	2413	Abbey	Murrow	Male	27	194 08
5	658	Donn	Bonnell	Male	38	194 01-
6	1243	Robbert	Blakey	Male	73	195 C
7	1565	Jay	Janiszewski	Male	71	195 08-
8	1177	Bobbette	Pozzi	Female	47	195 08-
9	3471	Brita	Afonso	Female	95	195 C

In [196…

```
df3["owns_car"].value_counts()
```

Out[196]:

```
Yes    2024
No     1976
Name: owns_car, dtype: int64
```

In [197…

```
df3["tenure"].value_counts()
```

Out[197]:

```
7.0     235
5.0     228
11.0    221
10.0    218
16.0    215
8.0     211
18.0    208
12.0    202
14.0    200
9.0     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
```

In [198…

```
# Investigating the last sheet of CustomerAddress
df4.head(10)
```

Out[198]:

	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
5	7	4 Delaware Trail	2210	New South Wales	Australia	9
6	8	49 Londonderry Lane	2650	New South Wales	Australia	4
7	9	97736 7th Trail	2023	New South Wales	Australia	12
8	11	93405 Ludington Park	3044	VIC	Australia	8
9	12	44339 Golden Leaf Alley	4557	QLD	Australia	4

In [199…

```
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 [200…

```
df4.isnull().sum()
```

Out[200]:

```
customer_id           0
address               0
postcode              0
state                 0
country               0
property_valuation    0
dtype: int64
```

This is good. There are no missing values in any of the columns.

In [201…

```
df4.duplicated().sum()
```

Out[201]:

```
0
```

This is good. There are no dupliated rows in this sheet.

In [202…

```
df4.nunique()
```

Out[202]:

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

In [203…

```
df4.shape
```

Out[203]:

```
(3999, 6)
```

In [204…

```
df4.columns
```

Out[204]:

```
Index(['customer_id', 'address', 'postcode', 'state', 'country',
       'property_valuation'],
      dtype='object')
```

In [205…

```
df4["address"].value_counts()
```

Out[205]:

```
3 Mariners Cove Terrace      2
3 Talisman Place             2
64 Macpherson Junction       2
359 Briar Crest Road         1
4543 Service Terrace         1
                            ..
5063 Shopko Pass             1
09 Hagan Pass                1
87897 Lighthouse Bay Pass    1
294 Lawn Junction            1
320 Acker Drive              1
Name: address, Length: 3996, dtype: int64
```

In [206…

```
df4["postcode"].value_counts()
```

Out[206]:

```
2170    31
2155    30
2145    30
2153    29
3977    26
        ..
3808     1
3114     1
4721     1
4799     1
3089     1
Name: postcode, Length: 873, dtype: int64
```

In [207…

```
df4["state"].value_counts()
```

Out[207]:

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

In [208…

```
df4["country"].value_counts()
```

Out[208]:

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

In [209…

```
df4["property_valuation"].value_counts()
```

Out[209]:

```
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
```

The values in the sheet seems proper and correct by investigating the columns.

In [230…

```
# Create a new Excel writer and add the four updated sheets
with pd.ExcelWriter("Downloads/new.xlsx") as writer:
    df1.to_excel(writer, sheet_name="Transactions", index=False)
    df2.to_excel(writer, sheet_name="NewCustomerList", index=False)
    df3.to_excel(writer, sheet_name="CustomerDemographic", index=False)
    df4.to_excel(writer, sheet_name="CustomerAddress", index=False)
```

By inputing a proper command in terminal, we can get and open the new update excel file in excel-reading applications.