Ajay Venkatesh Pandravada

Fetch Rewards Coding Exercise - Data Analyst

EDA and Data Quality Issues

This Document explores and analyzes each data source thoroughly and tries to identify the data quality issues that exist in them.

User Table:

```
[48] user_table.head()
```

	user_id	active	createdDate	lastLogin	role	signUpSource	state
0	5ff1e194b6a9d73a3a9f1052	True	2021-01-03 15:24:04	2021-01-03 15:25:37	consumer	Email	WI
1	5ff1e194b6a9d73a3a9f1052	True	2021-01-03 15:24:04	2021-01-03 15:25:37	consumer	Email	WI
2	5ff1e194b6a9d73a3a9f1052	True	2021-01-03 15:24:04	2021-01-03 15:25:37	consumer	Email	WI
3	5ff1e1eacfcf6c399c274ae6	True	2021-01-03 15:25:30	2021-01-03 15:25:30	consumer	Email	WI
4	5ff1e194b6a9d73a3a9f1052	True	2021-01-03 15:24:04	2021-01-03 15:25:37	consumer	Email	WI

```
[49] user_table.dtypes

user_id                object
active                   bool
createdDate    datetime64[ns]
lastLogin      datetime64[ns]
role                   object
signUpSource           object
state                  object
dtype: object
```

```
[50] user_table.isna().sum()

user_id          0
active           0
createdDate      0
lastLogin       62
role             0
signUpSource    48
state           56
dtype: int64
```

```
[51] user_table.user_id.nunique()

212
```

- The user table has 495 rows out of which only 212 unique user_ids can be seen.
  Based on the given columns, it can be assumed that either they are duplicates or users have logged in multiple times (in this case, a correlation can be found).

```
[52] user_table.groupby(['user_id','createdDate','lastLogin'],as_index=False)['role'].count().rename(
        columns={'role':'counts'}).sort_values(ascending=False, by = 'counts')
```

	user_id	createdDate	lastLogin	counts
0	54943462e4b07e684157a532	2014-12-19 14:21:22	2021-03-05 16:52:23	20
9	5fc961c3b8cfca11a077dd33	2020-12-03 22:08:03	2021-02-26 22:39:16	20
7	5fa41775898c7a11a6bcef3e	2020-11-05 15:17:09	2021-03-04 16:02:02	18
32	5ff5d15aeb7c7d12096d91a2	2021-01-06 15:03:54	2021-01-06 15:08:10	18
3	59c124bae4b0299e55b0f330	2017-09-19 14:07:54	2021-02-08 16:42:58	18
...	...	...	...	...
64	5ffca30604929111f6e92525	2021-01-11 19:12:06	2021-01-11 19:12:06	1
63	5ffc9d9f04929111f6e92456	2021-01-11 18:49:03	2021-01-11 18:49:03	1
61	5ffc9001b3348b11c93388b6	2021-01-11 17:50:57	2021-01-11 17:50:57	1
58	5ff8da7eb3348b11c9337b72	2021-01-08 22:19:42	2021-01-08 22:19:42	1
171	60268c7aefa6011bb1510786	2021-02-12 14:11:06	2021-02-12 14:11:06	1

- Lots of duplicate values for logins were identified, because of which it is not possible to determine if a user is logging in multiple times, which in turn affects the analysis of spending/rewards pattern.
  Hence, RFME analysis cannot be performed as it will be biased.

```
[53] user_tableDups = user_table[user_table.duplicated()]

[54] user_tableDups
```

	user_id	active	createdDate	lastLogin	role	signUpSource	state
1	5ff1e194b6a9d73a3a9f1052	True	2021-01-03 15:24:04	2021-01-03 15:25:37	consumer	Email	WI
2	5ff1e194b6a9d73a3a9f1052	True	2021-01-03 15:24:04	2021-01-03 15:25:37	consumer	Email	WI
4	5ff1e194b6a9d73a3a9f1052	True	2021-01-03 15:24:04	2021-01-03 15:25:37	consumer	Email	WI
5	5ff1e194b6a9d73a3a9f1052	True	2021-01-03 15:24:04	2021-01-03 15:25:37	consumer	Email	WI
8	5ff1e194b6a9d73a3a9f1052	True	2021-01-03 15:24:04	2021-01-03 15:25:37	consumer	Email	WI
...	...	...	...	...	...	...	...
490	54943462e4b07e684157a532	True	2014-12-19 14:21:22	2021-03-05 16:52:23	fetch-staff	NaN	NaN
491	54943462e4b07e684157a532	True	2014-12-19 14:21:22	2021-03-05 16:52:23	fetch-staff	NaN	NaN
492	54943462e4b07e684157a532	True	2014-12-19 14:21:22	2021-03-05 16:52:23	fetch-staff	NaN	NaN
493	54943462e4b07e684157a532	True	2014-12-19 14:21:22	2021-03-05 16:52:23	fetch-staff	NaN	NaN
494	54943462e4b07e684157a532	True	2014-12-19 14:21:22	2021-03-05 16:52:23	fetch-staff	NaN	NaN

283 rows × 7 columns

We can see there 283 duplicate rows. Let's drop the duplicates and analyze further.

```
[56] users = user_table.drop_duplicates(keep='first')

[57] users.isna().sum()

user_id          0
active           0
createdDate      0
lastLogin       40
role             0
signUpSource     5
state            6
dtype: int64
```

- Data Distribution:

```
[59] users.active.value_counts()

True     211
False      1
Name: active, dtype: int64
```

```
[61] users.signUpSource.value_counts()

Email     204
Google      3
Name: signUpSource, dtype: int64
```

```
[60] users.state.value_counts()

WI    193
AL      5
IL      3
KY      1
CO      1
OH      1
SC      1
NH      1
Name: state, dtype: int64
```

- From the data distribution it can be noticed that only one user is inactive and **91.47%** of the users are from Wisconsin, with no record for 6 users. **98.5%** of users signed using Email, and there is no data for 5 users.

Let’s see the distribution of number of users that joined the platform over months.

```
[63] grouped_data = users.groupby(['year', 'month'])['user_id'].count()

[64] grouped_data

year  month
2014  12        1
2015  4         1
2017  7         1
      9         1
      12        1
2020  1         1
      7         1
      11        4
      12        1
2021  1       170
      2        30
Name: user_id, dtype: int64
```

- We can observe that in 2021 january most of the users joined the platform.

Rewards Table:

```
[68] rewards_table.head()
```

	receipt_id	bonusPointsEarned	bonusPointsEarnedReason	createDate	dateScanned	finishedDate	modifyDate	pointsAwardedDate	poi
0	5ff1e1eb0a720f0523000575	500.0	Receipt number 2 completed, bonus point schedu...	2021-01-03 15:25:31	2021-01-03 15:25:31	2021-01-03 15:25:31	2021-01-03 15:25:36	2021-01-03 15:25:31	
1	5ff1e1bb0a720f052300056b	150.0	Receipt number 5 completed, bonus point schedu...	2021-01-03 15:24:43	2021-01-03 15:24:43	2021-01-03 15:24:43	2021-01-03 15:24:48	2021-01-03 15:24:43	
2	5ff1e1f10a720f052300057a	5.0	All-receipts receipt bonus	2021-01-03 15:25:37	2021-01-03 15:25:37	NaT	2021-01-03 15:25:42	NaT	
3	5ff1e1ee0a7214ada100056f	5.0	All-receipts receipt bonus	2021-01-03 15:25:34	2021-01-03 15:25:34	2021-01-03 15:25:34	2021-01-03 15:25:39	2021-01-03 15:25:34	
4	5ff1e1d20a7214ada1000561	5.0	All-receipts receipt bonus	2021-01-03 15:25:06	2021-01-03 15:25:06	2021-01-03 15:25:11	2021-01-03 15:25:11	2021-01-03 15:25:06	

```
[69] rewards_table.dtypes

receipt_id                    object
bonusPointsEarned            float64
bonusPointsEarnedReason       object
createDate            datetime64[ns]
dateScanned           datetime64[ns]
finishedDate          datetime64[ns]
modifyDate            datetime64[ns]
pointsAwardedDate     datetime64[ns]
pointsEarned                 float64
purchaseDate          datetime64[ns]
purchasedItemCount           float64
rewardsReceiptStatus          object
totalSpent                   float64
userId                        object
dtype: object
```

```
[70] rewards_table.isna().sum()

receipt_id              0
bonusPointsEarned     575
bonusPointsEarnedReason 575
createDate              0
dateScanned             0
finishedDate          551
modifyDate              0
pointsAwardedDate     582
pointsEarned          510
purchaseDate          448
purchasedItemCount    484
rewardsReceiptStatus    0
totalSpent            435
userId                  0
dtype: int64
```

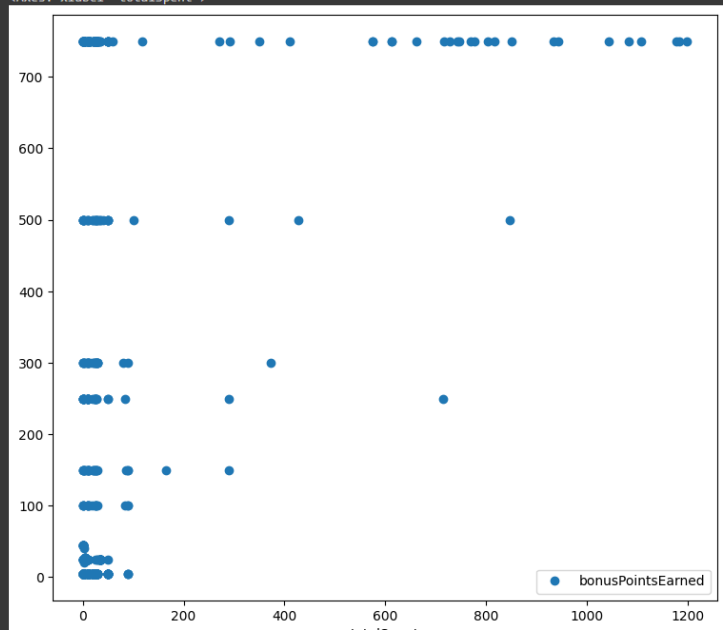
```
[71] rewards_table.receipt_id.nunique()

1119
```

```
[72] x = rewards_table["bonusPointsEarned"] - rewards_table["pointsEarned"]

[73] x

0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
       ...
1114    0.0
1115    NaN
1116    NaN
1117    0.0
1118    NaN
Length: 1119, dtype: float64
```

- All records in the rewards table are unique without any duplicates. But there are same columns with different names such as “bonusPointsEarned” and “pointsEarned”, which if avoided is better as they might create redundancies.
- Approximately 50% of the values are missing for the mentioned variables, which primarily consist of entries without barcodes or product purchases. These entries are not associated with rewards or user flags either. This situation warrants closer examination as it may indicate potential issues, such as unrecognized data or system glitches, which need to be addressed and resolved for accurate data analysis.

- Checking if there is a co-relation between total spent and bonus points earned.

```
[74] xx = rewards_table[rewards_table['totalSpent'] < 1500 ]

[75] xx.plot(x= 'totalSpent', y="bonusPointsEarned", style="o", figsize=(9, 8))

<Axes: xlabel='totalSpent'>
```



From the above graph we can see there isn't a strong relationship between the amount spent and bonusPointsEarned.

Category Table:

```
[76] category_table.head()
```

	category	categoryCode
0	Baking	BAKING
1	Beverages	BEVERAGES
2	Baking	BAKING
3	Baking	BAKING
4	Candy & Sweets	CANDY_AND_SWEETS

```
[77] category_table.category.value_counts()

Baking                          369
Beer Wine Spirits                90
Snacks                           75
Candy & Sweets                   71
Beverages                        63
Magazines                        44
Health & Wellness                44
Breakfast & Cereal               40
Grocery                          39
Dairy                            33
Condiments & Sauces              27
Frozen                           24
Personal Care                    20
Baby                             18
Canned Goods & Soups             12
Beauty                            9
Cleaning & Home Improvement       6
Deli                              6
Beauty & Personal Care            6
Household                         5
Bread & Bakery                    5
Dairy & Refrigerated              5
Outdoor                           1
Name: category, dtype: int64
```

```
[78] category_table.categoryCode.value_counts()

BAKING                          359
CANDY_AND_SWEETS                 71
BEER_WINE_SPIRITS                31
HEALTHY_AND_WELLNESS             14
GROCERY                          11
BABY                              7
CLEANING_AND_HOME_IMPROVEMENT     6
BREAD_AND_BAKERY                  5
DAIRY_AND_REFRIGERATED            5
PERSONAL_CARE                     4
BEVERAGES                         1
OUTDOOR                           1
MAGAZINES                         1
FROZEN                            1
Name: categoryCode, dtype: int64
```

```
[79] category_table.isna().sum()

category        155
categoryCode    650
dtype: int64
```

- There are a few categories which are not mapped to the category code and have NA value. This should be resolved immediately as it'll cause difficulties while promoting or analyzing the performance of a particular category over the others.

Ajay Venkatesh Pandravada

Items Table:

```
[80] items_table.head()
```

	barcode	description	finalPrice	itemPrice	needsFetchReview	partnerItemId	preventTargetGapPoints	quantityPurchased	userFlaggedBarcode	userFlaggedNewItem	...	itemNumber
0	4011	ITEM NOT FOUND	26.00	26.00	False	1	True	5.0	4011	True	...	NaN
1	4011	ITEM NOT FOUND	1	1	NaN	1	NaN	1.0	NaN	NaN	...	NaN
2	028400642255	DORITOS TORTILLA CHIP SPICY SWEET CHILI REDUCE...	10.00	10.00	True	2	True	1.0	028400642255	True	...	NaN
3	NaN	NaN	NaN	NaN	False	1	True	NaN	4011	True	...	NaN
4	4011	ITEM NOT FOUND	28.00	28.00	False	1	True	4.0	4011	True	...	NaN

- There are many columns which have more than 50% of data missing, the data collected have entries which have no barcode and most of the review related columns have many NA values.
  This can be avoided if while collecting the data review data and item data are separated. Later if needed can be joined using the receipt_id as the common column.
- Let's consider a few important columns and draw some insights from the data.

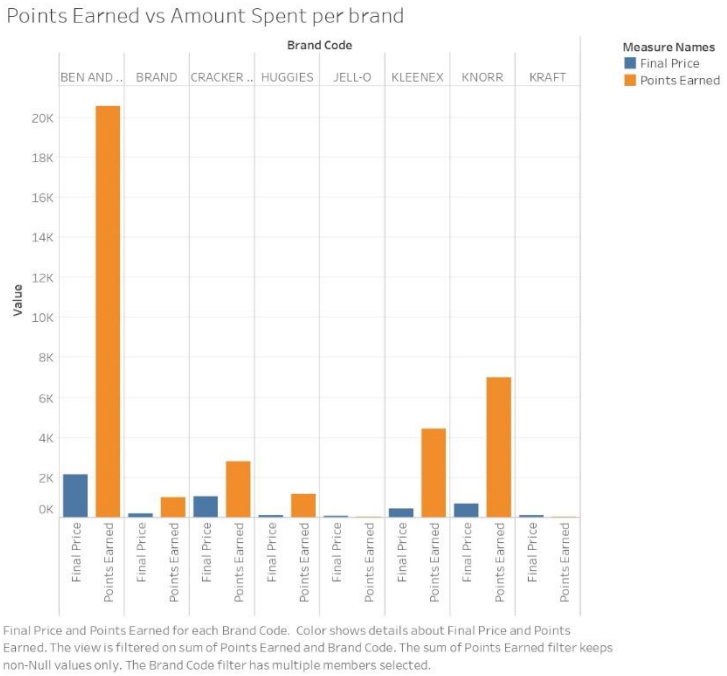
```
[81] items_table.isna().sum()
```

```
barcode                                3851
description                             381
finalPrice                              174
itemPrice                               174
needsFetchReview                       6128
partnerItemId                             0
preventTargetGapPoints                 6583
quantityPurchased                       174
userFlaggedBarcode                     6604
userFlaggedNewItem                     6618
userFlaggedPrice                       6642
userFlaggedQuantity                    6642
receipt_id                                0
needsFetchReviewReason                 6722
pointsNotAwardedReason                 6601
pointsPayerId                          5674
rewardsGroup                           5210
rewardsProductPartnerId                4672
userFlaggedDescription                 6736
originalMetaBriteBarcode               6870
originalMetaBriteDescription           6931
brandCode                              4341
competitorRewardsGroup                 6666
discountedItemPrice                    1172
originalReceiptItemText                1181
itemNumber                             6788
originalMetaBriteQuantityPurchased     6926
pointsEarned                           6014
targetPrice                            6563
competitiveProduct                     6296
originalFinalPrice                     6932
originalMetaBriteItemPrice             6932
deleted                                6932
priceAfterCoupon                       5985
metabriteCampaignId                    6078
dtype: int64
```

```
[85] items_table1['pointsEarned'] = items_table1['pointsEarned'].astype(float)
     items_table1['finalPrice'] = items_table1['finalPrice'].astype(float)

[86] items_table1.groupby("brandCode").pointsEarned.sum().sort_values(ascending = False).head()

     brandCode
     BEN AND JERRYS    20548.6
     KNORR              6993.5
     KLEENEX            4440.3
     CRACKER BARREL     2814.0
     HUGGIES            1199.4
     Name: pointsEarned, dtype: float64

[87] items_table1.groupby("brandCode").finalPrice.sum().sort_values(ascending = False).head(5)

     brandCode
     BEN AND JERRYS    2149.45
     HEMPLER'S         1102.30
     CRACKER BARREL    1069.32
     KNORR              706.11
     HY-VEE             656.62
     Name: finalPrice, dtype: float64
```

Ajay Venkatesh Pandravada

Using Tableau, the following analysis was performed:



Final Price and Points Earned for each Brand Code. Color shows details about Final Price and Points Earned. The view is filtered on sum of Points Earned and Brand Code. The sum of Points Earned filter keeps non-Null values only. The Brand Code filter has multiple members selected.

Brands Table:

```
[89] brand_table.head()
```

	brand_id	barcode	category	categoryCode	name	topBrand	brandCode	cpg.id	cpg.ref
0	601ac115be37ce2ead437551	511111019862	Baking	BAKING	test brand @1612366101024	0.0	NaN	601ac114be37ce2ead437550	Cogs
1	601c5460be37ce2ead43755f	511111519928	Beverages	BEVERAGES	Starbucks	0.0	STARBUCKS	5332f5fbe4b03c9a25efd0ba	Cogs
2	601ac142be37ce2ead43755d	511111819905	Baking	BAKING	test brand @1612366146176	0.0	TEST BRANDCODE @1612366146176	601ac142be37ce2ead437559	Cogs
3	601ac142be37ce2ead43755a	511111519874	Baking	BAKING	test brand @1612366146051	0.0	TEST BRANDCODE @1612366146051	601ac142be37ce2ead437559	Cogs
4	601ac142be37ce2ead43755e	511111319917	Candy & Sweets	CANDY_AND_SWEETS	test brand @1612366146827	0.0	TEST BRANDCODE @1612366146827	5332fa12e4b03c9a25efd1e7	Cogs

```
[90] brand_table.dtypes

brand_id          object
barcode            int64
category          object
categoryCode      object
name              object
topBrand         float64
brandCode         object
cpg.id            object
cpg.ref           object
dtype: object
```

```
[91] brand_table.isna().sum()

brand_id            0
barcode             0
category          155
categoryCode      650
name                0
topBrand          612
brandCode         234
cpg.id              0
cpg.ref             0
dtype: int64
```

```
[92] brand_table.brandCode.value_counts()

                                    35
GOODNITES                            2
HUGGIES                              2
ROYAL DANSK                          1
SOL                                  1
                                    ..
TEST BRANDCODE @1599159969028        1
TEST BRANDCODE @1597350074404        1
SEDAL                                1
RED ROCK DELI                        1
TEST BRANDCODE @1613158231644        1
Name: brandCode, Length: 897, dtype: int64
```

- Data is not mapped properly leading to many barcodes not falling under any brand and there are few anomalies, for example, an entry with brand named STARBUCKS has brandcode: Starbacks in one and brandcode: beverage in another, which will cause issues while analyzing performance of brands. Having brand code will immensely help to identify top performing brand and their competitors.

```
[95] brand_table.groupby(['categoryCode', 'brandCode'])['barcode'].count()

     categoryCode   brandCode
     BABY           A+D                 1
                    GERBER BABY FOOD    1
                    GERBER GOOD START   1
                    GOODNITES           1
                    HUGGIES             1
                                       ..
     OUTDOOR        KINGSFORD           1
     PERSONAL_CARE  DEGREE              1
                    DEPEND              1
                    TRESEMME            1
                    VASELINE            1
     Name: barcode, Length: 444, dtype: int64
```

- An attempt has been made to identify top brands in each category but due to inconsistencies in data the insights are not effective.