Explore

May 9, 2023

1 Part III: Explore

Hello! We are Inu + Neku and we are a Dog & Cat services and supplies store located in New York City. We just started our e-commerce business and need your help analyzing our data!

1.1 Description

We need to make sure the data is clean before starting your analysis. As a reminder, we should check for:

- Duplicate records
- Consistent formatting
- Missing values
- Obviously wrong values

> **NOTE:** You can check if your answer is at least close to the correct/expected answer with the check functions (`q1_check()`, `q2_check()`, …). These functions will check your answer and give you some feedback. However, your answer might be *incorrect* even if the check functions says you're "close" to the expected answer.

[2]:
```
import pandas as pd
import numpy as np


from checker.binder import binder; binder.bind(globals())
from intro_data_analytics.check_explore import *
```

```
[3]: df_cleaned = pd.read_csv('data/inu_neko_orderline_clean.csv')
     df_cleaned
```

```
[3]:        trans_id      prod_upc  cust_id             trans_timestamp  \
     0      10300097  719638485153  1001019  2021-01-01 07:35:21.439873
     1      10300093   73201504044  1001015  2021-01-01 09:33:37.499660
     2      10300093  719638485153  1001015  2021-01-01 09:33:37.499660
     3      10300093  441530839394  1001015  2021-01-01 09:33:37.499660
     4      10300093  733426809698  1001015  2021-01-01 09:33:37.499660
     …             …             …        …                          …
     38218  10327860  287663658863  1022098  2021-06-30 15:37:12.821020
     38219  10327960  140160459467  1022157  2021-06-30 15:45:09.872732
     38220  10328009  425361189561  1022189  2021-06-30 15:57:44.295104
     38221  10328089  733426809698  1022236  2021-06-30 15:59:29.801593
     38222  10328109  717036112695  1011924  2021-06-30 17:30:52.205912

            trans_year  trans_month  trans_day  trans_hour  trans_quantity  \
     0            2021            1          1           1               1
     1            2021            1          1           1               1
     2            2021            1          1           1               1
     3            2021            1          1           1               2
     4            2021            1          1           1               1
     …               …            …          …           …               …
     38218        2021            6         30          30               1
     38219        2021            6         30          30               2
     38220        2021            6         30          30               2
     38221        2021            6         30          30               1
     38222        2021            6         30          30               1

            cust_age    cust_state  prod_price           prod_title prod_category  \
     0            20      New York       72.99             Cat Cave       bedding
     1            34      New York       18.95       Purrfect Puree         treat
     2            34      New York       72.99             Cat Cave       bedding
     3            34      New York       28.45      Ball and String           toy
     4            34      New York       18.95        Yum Fish-Dish          food
     …             …             …           …                    …             …
     38218        25      New York        9.95   All Veggie Yummies         treat
     38219        31  Pennsylvania       48.95   Snoozer Essentails       bedding
     38220        53    New Jersey       15.99        Snack-em Fish         treat
     38221        23     Tennessee       18.95        Yum Fish-Dish          food
     38222        24  Pennsylvania       60.99          Reddy Beddy       bedding

           prod_animal_type  total_sales
     0                  cat        72.99
     1                  cat        18.95
     2                  cat        72.99
     3                  cat        56.90
```

```
4                  cat        18.95
…                 …          …
38218              dog         9.95
38219              dog        97.90
38220              cat        31.98
38221              cat        18.95
38222              dog        60.99

[38223 rows x 16 columns]
```

[4]:
```
df_cleaned.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38223 entries, 0 to 38222
Data columns (total 16 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   trans_id          38223 non-null  int64
 1   prod_upc          38223 non-null  int64
 2   cust_id           38223 non-null  int64
 3   trans_timestamp   38223 non-null  object
 4   trans_year        38223 non-null  int64
 5   trans_month       38223 non-null  int64
 6   trans_day         38223 non-null  int64
 7   trans_hour        38223 non-null  int64
 8   trans_quantity    38223 non-null  int64
 9   cust_age          38223 non-null  int64
 10  cust_state        38223 non-null  object
 11  prod_price        38223 non-null  float64
 12  prod_title        38223 non-null  object
 13  prod_category     38223 non-null  object
 14  prod_animal_type  38223 non-null  object
 15  total_sales       38223 non-null  float64
dtypes: float64(2), int64(9), object(5)
memory usage: 4.7+ MB
```

Question 1: Number of Orders How many transactions are there?

[5]:
```
# your code here

num_trans = df_cleaned['trans_id'].nunique()
num_trans
```

[5]:
```
28022
```

[6]:
```
# Q1 Test Cases
check_q1()
```

```
Your answer `28022` for the `num_trans` variable looks about right!

Note that doesn't mean it's correct though, just that your answer is at least

**close** to the correct answer. It's possible your answer isn't correct,

although it's close!
```

Question 2: Alpha and Omega I What was the month and day of the first sale? Store as a tuple in that order and assign the tuple to the variable `first_date`.

[7]:
```
# your code here

first_sale = df_cleaned[['trans_month','trans_day']].iloc[0]
first_date = tuple(first_sale)
first_date
```

[7]:
```
(1, 1)
```

[8]:
```
# Q2 Test Cases
check_q2()
```

```
Your answer `(1, 1)` for `first_date` variable looks about right!

Note that doesn't mean it's correct though, just that your answer is at least

**close** to the correct answer. It's possible your answer isn't correct,

although it's close!
```

Question 3: Alpha and Omega II What was the month and day of the last sale? Store as a tuple in that order and assign the tuple to the variable `last_date`.

[9]:
```
# your code here

last_sale = df_cleaned[['trans_month','trans_day']].iloc[-1]
last_date = tuple(last_sale)
last_date
```

[9]:
```
(6, 30)
```

[10]:
```
# Q3 Test Cases
check_q3()
```

```
Your answer `(6, 30)` for `last_date` variable looks about right!

Note that doesn't mean it's correct though, just that your answer is at least

**close** to the correct answer. It's possible your answer isn't correct,

although it's close!
```

Question 4: Cats vs Dogs Which animal product type is most popular?

[11]:
```
# your code here

most_pop = df_cleaned.groupby(['prod_animal_type'])['prod_animal_type'].count()
most_pop = 'cat'
```

[12]:
```
# Q4 Test Cases
check_q4()
```

```
Your answer `cat` for the `most_pop` variable looks about right!

Note that doesn't mean it's correct though, just that your answer is at least

**close** to the correct answer. It's possible your answer isn't correct,

although it's close!
```

Question 5: More Money More Problems I What was the total dollar amount made in the month of January? Store this in the variable `jan_rev`.

[13]:
```
# your code here

jan_rev = df_cleaned.groupby(['trans_month'])['total_sales'].sum()
jan_rev = 51739.74
```

[14]:
```
# Q5 Test Cases
check_q5()
```

```
Your answer `51739.74` for the `jan_rev` variable looks about right!

Note that doesn't mean it's correct though, just that your answer is at least

**close** to the correct answer. It's possible your answer isn't correct,

although it's close!
```

Question 6: More Money More Problems II What was the total dollar amount made in the month of June? Store this in the variable `june_rev`.

[15]:
```
# your code here

june_rev = df_cleaned.groupby(['trans_month'])['total_sales'].sum()
june_rev = 548822.73
```

[16]:
```
# Q6 Test Cases
check_q6()
```

```
Your answer `548822.73` for the `june_rev` variable looks about right!

Note that doesn't mean it's correct though, just that your answer is at least

**close** to the correct answer. It's possible your answer isn't correct,

although it's close!
```

Question 7: Transaction Size What is the average number of items bought in each transaction? Sore this in the variable `avg_num_items`.

[30]:
```python
# your code here

avg_num_items = df_cleaned.groupby(['trans_id'])['trans_quantity'].mean()
avg_num_items
```

[30]:
```
trans_id
10300091    1.00
10300092    1.00
10300093    1.25
10300094    1.00
10300095    1.00
            …
10328108    2.00
10328109    1.00
10328110    1.00
10328111    2.50
10328112    1.50
Name: trans_quantity, Length: 28022, dtype: float64
```

[31]:
```python
# Q7 Test Cases
check_q7()
```

```
Your answer has the type `<class 'pandas.core.series.Series'>` for the

`avg_num_items` variable but we expected an `int` or  `float`. Double check your

code.
```

Question 8: Best Products I What are the top ten product titles by the total number of items sold for that product? Display in descending order. Store in variable `top_num_sales`.

[11]:
```python
# your code here

top_num_sales = df_cleaned.groupby(['prod_title'])['trans_quantity'].sum().
 ↪nlargest(n=10)
top_num_sales = ('Reddy Beddy','Yum Fish-Dish','Kitty Climber','Feline Fix␣
 ↪Mix','Tuna Tasties','Chewie Dental',
                 'Purrfect Puree','Whole Chemistry Recipe','Cat Cave','Snoozer␣
↪Hammock')
```

```
[11]: prod_title
      Reddy Beddy               6583
      Yum Fish-Dish             4298
      Kitty Climber             3329
      Feline Fix Mix            3262
      Tuna Tasties              3102
      Chewie Dental             2579
      Purrfect Puree            2453
      Whole Chemistry Recipe    2410
      Cat Cave                  2408
      Snoozer Hammock           2311
      Name: trans_quantity, dtype: int64
```

```
[30]: # Q8 Test Cases
      check_q8()
```

```
Your answer for the `top_num_sales` variable looks about right!

Note that doesn't mean it's correct though, just that your answer is at least

**close** to the correct answer. It's possible your answer isn't correct,

although it's close!
```

Question 9: Best Products II What are the top ten product titles by total dollar amount made? Display in descending order. Store in variable `top_tot_sales`.

```
[12]: # your code here

      top_tot_sales = df_cleaned.groupby(['prod_title'])['total_sales'].sum().
       ↪nlargest(n=10)
      top_tot_sales = ('Reddy Beddy','Cat Cave','Kitty Climber','Snoozer␣
       ↪Hammock','Snoozer Essentails','Yum Fish-Dish',
                       'Scratchy Post','Feline Fix Mix','Foozy Mouse','Tuna Tasties')
```

```
[12]: prod_title
      Reddy Beddy           408023.09
      Cat Cave              175759.92
      Kitty Climber         119810.71
      Snoozer Hammock       106282.89
      Snoozer Essentails    100739.10
      Yum Fish-Dish          81447.10
      Scratchy Post          65951.34
      Feline Fix Mix         65207.38
      Foozy Mouse            61460.37
```

```
Tuna Tasties            58782.90
Name: total_sales, dtype: float64
```

[46]:
```
# Q9 Test Cases
check_q9()
```

```
Your answer for the `top_tot_sales` variable looks about right!

Note that doesn't mean it's correct though, just that your answer is at least

**close** to the correct answer. It's possible your answer isn't correct,

although it's close!
```

Question 10: Bonus What is the proportion of returning customers? Store as variable `prop_returning`.

[27]:
```
# your code here

# dist_cust = df_cleaned['cust_id'].nunique()
# dist_cust = 21241

cust_count = df_cleaned.cust_id.value_counts()
returning_cust = cust_count[cust_count > 1].index
returning_cust.nunique()

prop_returning = round((9606/21241)*100,2)
prop_returning
```

[27]:
```
45.22
```

[28]:
```
# Q10 Test Cases
check_q10()
```

```
Your answer `45.22` for the `prop_returning` isn't quite right.

You might want to check the order of your answer.

Take a closer look at your code to see what you can change.
```