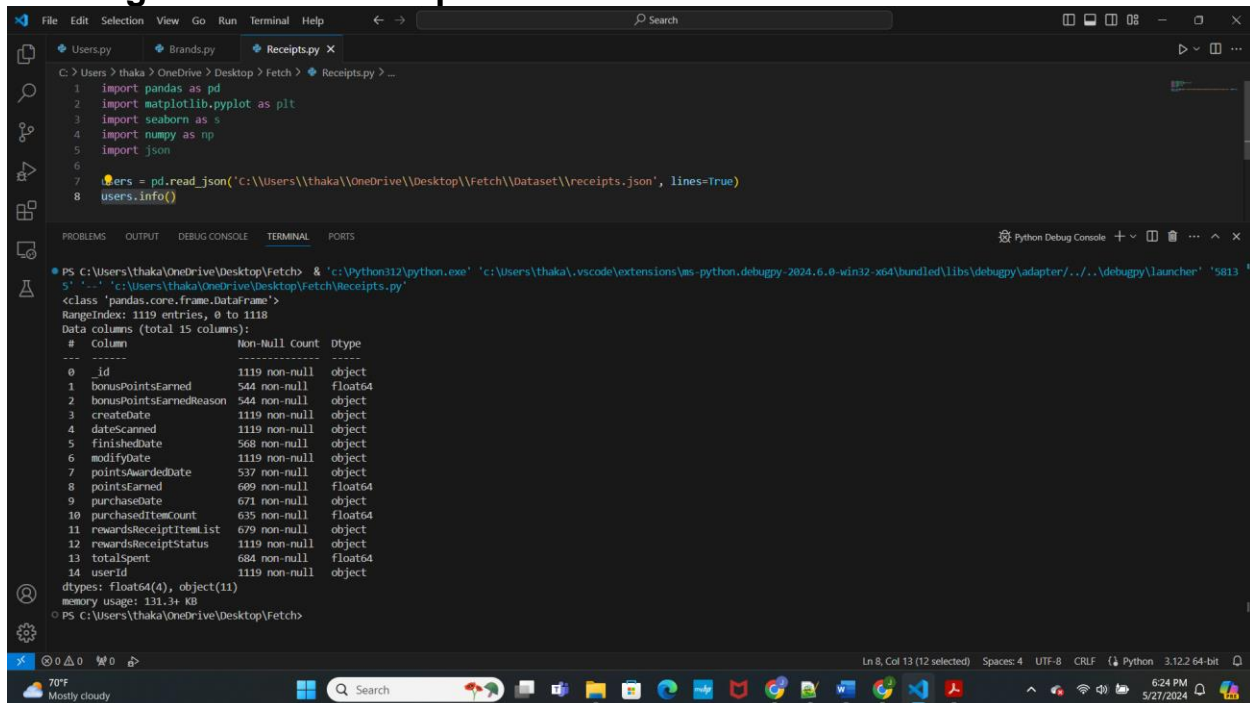


Loading dataset for receipts



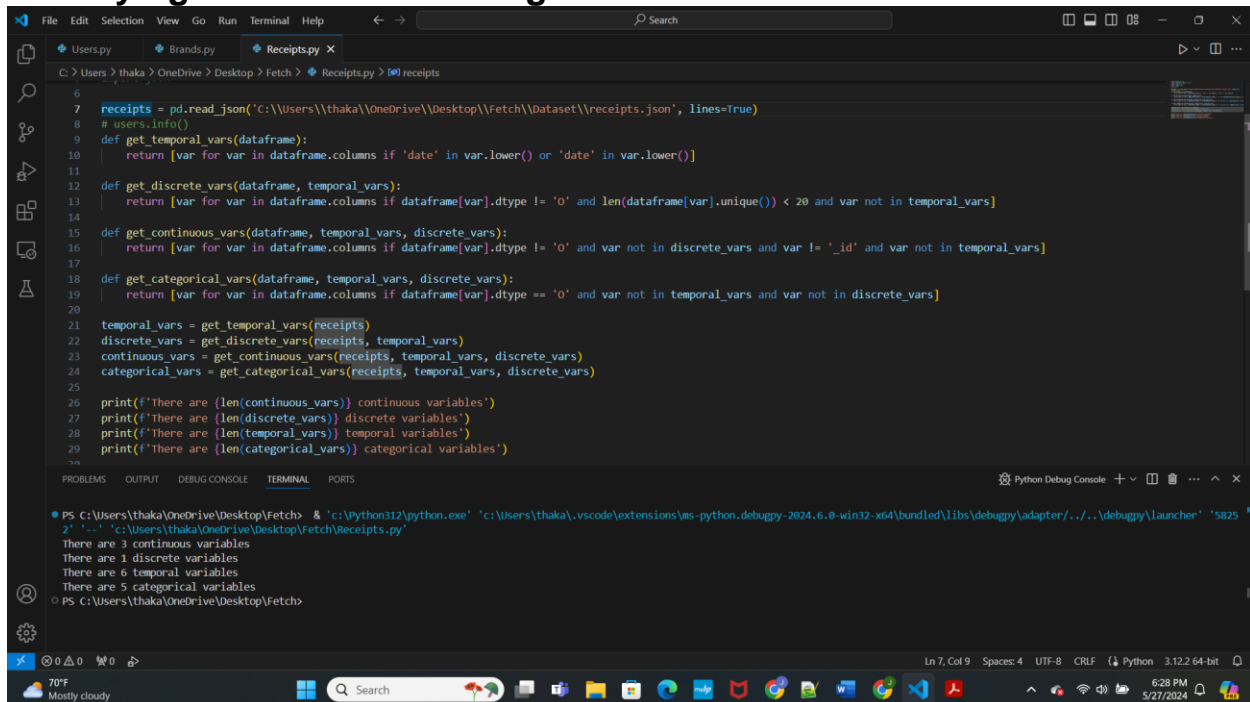
The screenshot shows a VS Code editor with a file named `Receipts.py` open. The code imports `pandas`, `matplotlib.pyplot`, `seaborn`, `numpy`, and `json`. It then uses `pd.read_json` to load a dataset from a local file path. The output in the terminal shows the DataFrame's structure, including the number of entries, columns, and data types.

```
C:\Users\thaka> OneDrive > Desktop > Fetch > Receipts.py > ...
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as s
4 import numpy as np
5 import json
6
7 users = pd.read_json('c:\\Users\\thaka\\OneDrive\\Desktop\\Fetch\\dataset\\receipts.json', lines=True)
8 users.info()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python Debug Console

```
PS C:\Users\thaka\OneDrive\Desktop\Fetch> & 'c:\Python312\python.exe' 'c:\Users\thaka\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundle\libs\debugpy\adapter\...\debugpy\launcher' '5813'
5' '-' 'c:\Users\thaka\OneDrive\Desktop\Fetch\Receipts.py'
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1119 entries, 0 to 1118
Data columns (total 15 columns):
# Column Non-Null Count Dtype
---
0 _id 1119 non-null object
1 bonusPointsEarned 544 non-null float64
2 bonusPointsEarnedReason 544 non-null object
3 createDate 1119 non-null object
4 dateScanned 1119 non-null object
5 finishedDate 568 non-null object
6 modifyDate 1119 non-null object
7 pointsAwardedDate 537 non-null object
8 pointsEarned 609 non-null float64
9 purchaseDate 671 non-null object
10 purchasedItemCount 635 non-null float64
11 rewardsReceiptItemList 679 non-null object
12 rewardsReceiptStatus 1119 non-null object
13 totalSpent 684 non-null float64
14 userId 1119 non-null object
dtypes: float64(4), object(11)
memory usage: 131.3+ KB
PS C:\Users\thaka\OneDrive\Desktop\Fetch>
```

Identifying numerical and categorical variables

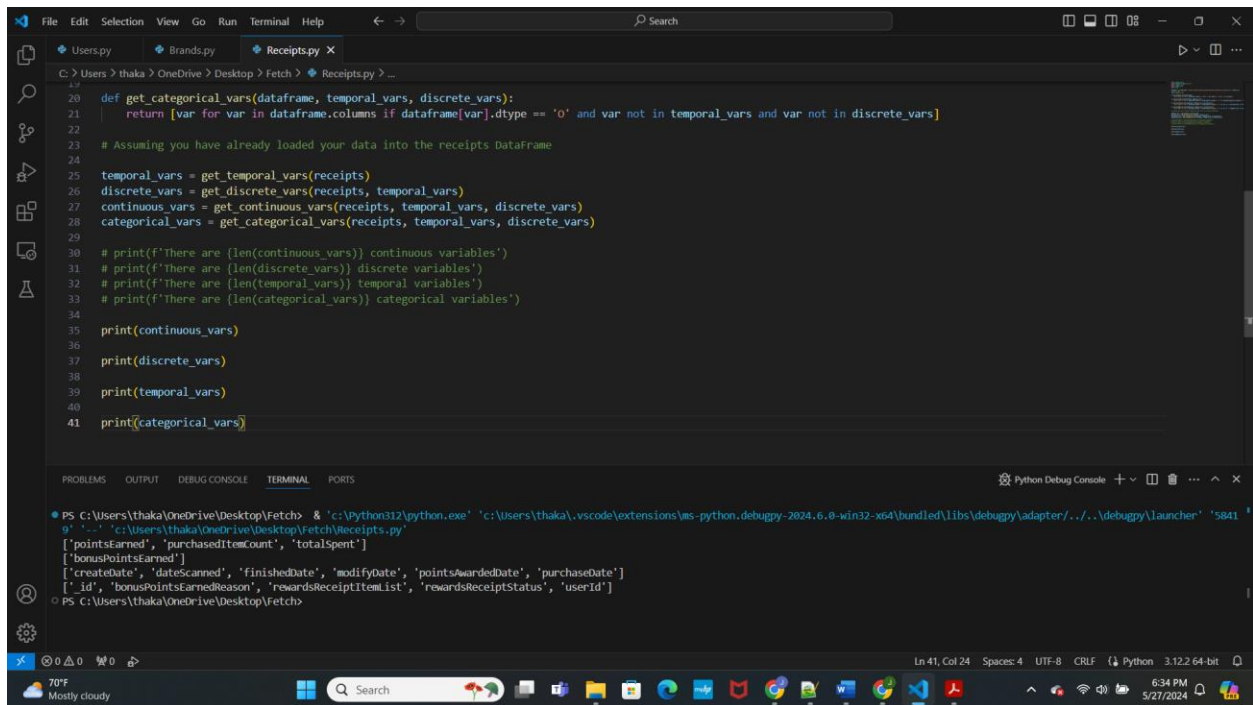


The screenshot shows the same VS Code editor with `Receipts.py` open. The code defines functions to identify temporal, discrete, continuous, and categorical variables in a DataFrame. It then applies these functions to the `receipts` DataFrame and prints the results.

```
C:\Users\thaka> OneDrive > Desktop > Fetch > Receipts.py > @ receipts
6
7 receipts = pd.read_json('c:\\Users\\thaka\\OneDrive\\Desktop\\Fetch\\dataset\\receipts.json', lines=True)
8 # users.info()
9 def get_temporal_vars(dataframe):
10     return [var for var in dataframe.columns if 'date' in var.lower() or 'date' in var.lower()]
11
12 def get_discrete_vars(dataframe, temporal_vars):
13     return [var for var in dataframe.columns if dataframe[var].dtype != 'o' and len(dataframe[var].unique()) < 20 and var not in temporal_vars]
14
15 def get_continuous_vars(dataframe, temporal_vars, discrete_vars):
16     return [var for var in dataframe.columns if dataframe[var].dtype != 'o' and var not in discrete_vars and var != '_id' and var not in temporal_vars]
17
18 def get_categorical_vars(dataframe, temporal_vars, discrete_vars):
19     return [var for var in dataframe.columns if dataframe[var].dtype == 'o' and var not in temporal_vars and var not in discrete_vars]
20
21 temporal_vars = get_temporal_vars(receipts)
22 discrete_vars = get_discrete_vars(receipts, temporal_vars)
23 continuous_vars = get_continuous_vars(receipts, temporal_vars, discrete_vars)
24 categorical_vars = get_categorical_vars(receipts, temporal_vars, discrete_vars)
25
26 print(f'There are {len(continuous_vars)} continuous variables')
27 print(f'There are {len(discrete_vars)} discrete variables')
28 print(f'There are {len(temporal_vars)} temporal variables')
29 print(f'There are {len(categorical_vars)} categorical variables')
30
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python Debug Console

```
PS C:\Users\thaka\OneDrive\Desktop\Fetch> & 'c:\Python312\python.exe' 'c:\Users\thaka\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundle\libs\debugpy\adapter\...\debugpy\launcher' '5825'
2' '-' 'c:\Users\thaka\OneDrive\Desktop\Fetch\Receipts.py'
There are 3 continuous variables
There are 1 discrete variables
There are 6 temporal variables
There are 5 categorical variables
PS C:\Users\thaka\OneDrive\Desktop\Fetch>
```



The screenshot shows a VS Code editor with a file named `Receipts.py`. The script defines functions to classify variables in a DataFrame as temporal, discrete, continuous, or categorical. It then prints the counts for each category. The terminal output shows the results of running the script.

```
def get_categorical_vars(dataframe, temporal_vars, discrete_vars):
    return [var for var in dataframe.columns if dataframe[var].dtype == 'O' and var not in temporal_vars and var not in discrete_vars]

# Assuming you have already loaded your data into the receipts DataFrame

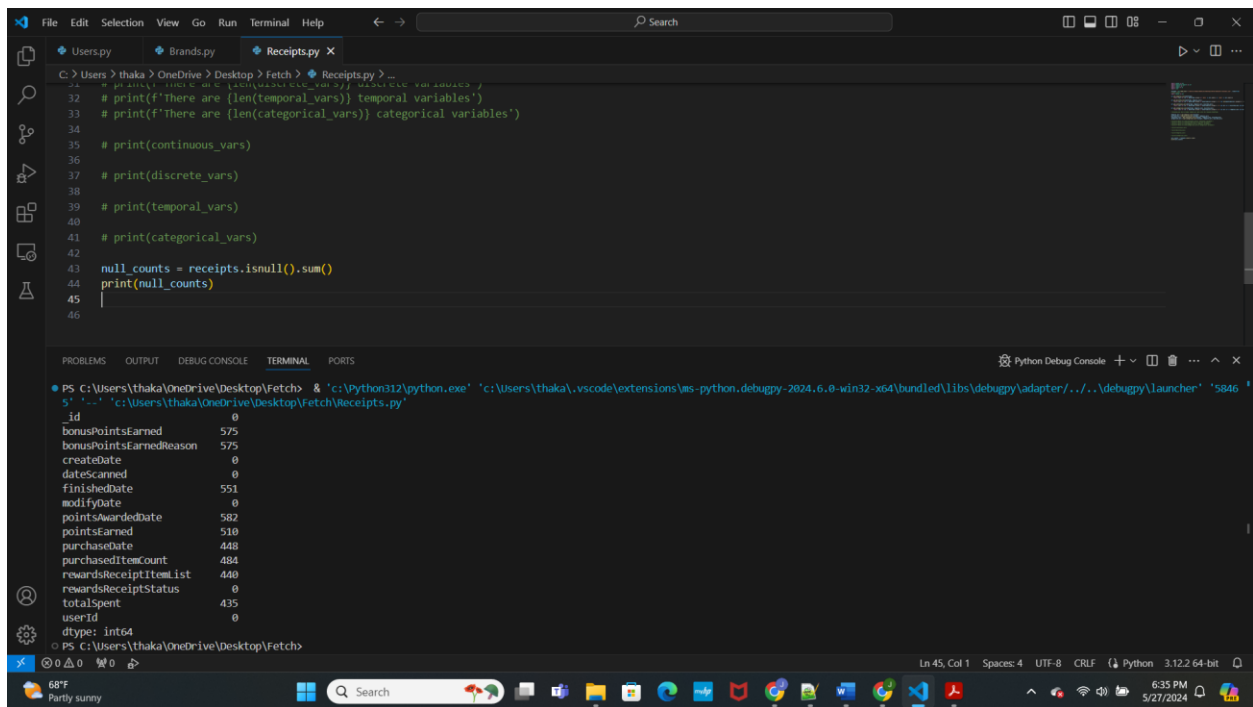
temporal_vars = get_temporal_vars(receipts)
discrete_vars = get_discrete_vars(receipts, temporal_vars)
continuous_vars = get_continuous_vars(receipts, temporal_vars, discrete_vars)
categorical_vars = get_categorical_vars(receipts, temporal_vars, discrete_vars)

# print(f'There are {len(continuous_vars)} continuous variables')
# print(f'There are {len(discrete_vars)} discrete variables')
# print(f'There are {len(temporal_vars)} temporal variables')
# print(f'There are {len(categorical_vars)} categorical variables')

print(continuous_vars)
print(discrete_vars)
print(temporal_vars)
print(categorical_vars)
```

```
PS C:\Users\thaka\OneDrive\Desktop\Fetch> & 'c:\Python312\python.exe' 'c:\Users\thaka\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundle\libs\debugpy\adapter\...\debugpy\launcher' '5841' '...' 'c:\Users\thaka\OneDrive\Desktop\Fetch\Receipts.py'
["pointsEarned", "purchasedItemCount", "totalSpent"]
["bonusPointsEarned"]
["createDate", "dateScanned", "finishedDate", "modifyDate", "pointsAwardedDate", "purchaseDate"]
["_id", "bonusPointsEarnedReason", "rewardsReceiptItemList", "rewardsReceiptStatus", "userId"]
PS C:\Users\thaka\OneDrive\Desktop\Fetch>
```

Quantifying missing data



The screenshot shows a VS Code editor with a file named `Receipts.py`. The script prints the counts for each variable category and then calculates the total number of null values in the DataFrame. The terminal output shows the results of running the script.

```
# print(f'There are {len(temporal_vars)} temporal variables')
# print(f'There are {len(categorical_vars)} categorical variables')

# print(continuous_vars)
# print(discrete_vars)
# print(temporal_vars)
# print(categorical_vars)

null_counts = receipts.isnull().sum()
print(null_counts)
```

```
PS C:\Users\thaka\OneDrive\Desktop\Fetch> & 'c:\Python312\python.exe' 'c:\Users\thaka\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundle\libs\debugpy\adapter\...\debugpy\launcher' '5846' '...' 'c:\Users\thaka\OneDrive\Desktop\Fetch\Receipts.py'
_id
bonusPointsEarned
bonusPointsEarnedReason
createDate
dateScanned
finishedDate
modifyDate
pointsAwardedDate
pointsEarned
purchaseDate
purchasedItemCount
rewardsReceiptItemList
rewardsReceiptStatus
totalSpent
userId
dtype: int64
PS C:\Users\thaka\OneDrive\Desktop\Fetch>
```

The screenshot shows a VS Code editor with a Python file named 'Receipts.py' open. The script is located at 'C:\Users\thaka\OneDrive\Desktop\Fetch\Receipts.py'. The code calculates the percentage of null values for each column in a dataset and iterates through the dictionary of column names and their corresponding null percentages. The terminal output shows the results of these calculations for a specific receipt item.

```

C:\Users\thaka\OneDrive\Desktop\Fetch> cd Receipts.py >
41 # print(categorical_vars)
42
43 # null_counts = receipts.isnull().sum()
44 # print(null_counts)
45
46 # Calculate the percentage of null values for each column
47 percentage_null_values = receipts.isnull().mean()
48
49 # Iterate through the dictionary of column names and their corresponding null percentages
50 for key, value in percentage_null_values.items():
51     # Check if the null percentage is greater than 0
52     if value > 0:
53         # Print the column name and its null percentage
54         print(key, ":", value * 100)
55
56
57

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python Debug Console

```

PS C:\Users\thaka\OneDrive\Desktop\Fetch> & 'c:\python312\python.exe' 'c:\Users\thaka\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundle\libs\debugpy\adapter\...\debugpy\launcher' '5849'
1' '-c' 'c:\Users\thaka\OneDrive\Desktop\Fetch\Receipts.py'
bonusPointsEarned : 51.385165326184094
bonusPointsEarnedReason : 51.385165326184094
finishedDate : 49.240393208221626
pointsAwardedDate : 52.01072386058981
pointsEarned : 45.57640750670242
purchaseDate : 40.03574620196604
purchaseItemCount : 43.25290437890974
rewardsReceiptItemList : 39.32082216264522
totalSpent : 38.8739946380607
PS C:\Users\thaka\OneDrive\Desktop\Fetch>

```

68°F Partly sunny

```
File Edit Selection View Go Run Terminal Help
C:\Users> thaka > OneDrive > Desktop > Fetch > Receipts.py > ...
53 # ... Print the column name and its null percentage
54 # print(key, ":", value * 100)
55 unique_points_earned = receipts["points_earned"].unique()
56 print(unique_points_earned)
57
58
59
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

Python Debug Console

```
PS C:\Users\thaka\OneDrive\Desktop\Fetch> & 'c:\Python312\python.exe' 'c:\Users\thaka\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundle\libs\debugpy\adapter\..\..\debugpy\launcher' '5850'
8' '-' 'c:\Users\thaka\OneDrive\Desktop\Fetch\Receipts.py'
[5.000000e+02 1.500000e+02 5.000000e+00 7.500000e+02 2.500000e+02 1.000000e+02
8.850000e+03 3.000000e+02 nan 3.892000e+02 1.850000e+02 3.500000e+01
6.500000e+02 5.500000e+01 5.000000e+01 3.550000e+02 6.000000e+02 1.750000e+03
3.500000e+02 2.250000e+02 2.750000e+02 2.500000e+01 7.550000e+02 1.800000e+03
8.100000e+02 3.050000e+02 9.450000e+03 9.120000e+01 8.250000e+02 3.500000e+02
1.250000e+02 7.910000e+02 2.000000e+02 2.250000e+02 8.000000e+00 4.805000e+03
2.055000e+02 8.412000e+02 5.750000e+03 3.750000e+03 8.700000e+03 7.600000e+02
7.800000e+02 9.200000e+03 1.005000e+03 1.999600e+03 1.892000e+02 8.950000e+03
8.850000e+02 8.000000e+02 2.950000e+02 6.824000e+02 8.374000e+02 2.378000e+02
1.600000e+02 8.557000e+02 6.057000e+02 2.416700e+03 1.806400e+03 4.057000e+02
1.516900e+03 1.658300e+03 2.685800e+03 8.791000e+02 3.659400e+03 9.344000e+02
8.777000e+02 9.221000e+02 1.541800e+03 1.000000e+03 5.744000e+02 5.060000e+01
2.055500e+03 5.850000e+03 4.850000e+03 1.736000e+02 9.850000e+03 5.090000e+01
2.300000e+03 6.730000e+02 4.059000e+02 2.143300e+03 1.550000e+03 9.865000e+02
5.834000e+02 4.480500e+03 3.379000e+03 3.236000e+02 6.257300e+03 2.407700e+03
1.178700e+03 1.447000e+03 1.725500e+03 1.476200e+03 1.788000e+03 1.077500e+03
1.135100e+03 1.044300e+03 6.407000e+02 5.236000e+02 4.877000e+02 7.137200e+03
1.205000e+02 4.000000e+02 1.019900e+04 8.500000e+02 9.460000e+01 1.850000e+03
8.400000e+02 1.049800e+03 9.400000e+02 4.944700e+03 3.750000e+02 1.499500e+03
2.099000e+02 2.098000e+02 2.100000e+02 2.095000e+02 7.892000e+02 3.500000e+03]
PS C:\Users\thaka\OneDrive\Desktop\Fetch>
```

Checking for redundant records

```
File Edit Selection View Go Run Terminal Help
C:\Users\thaka> OneDrive\Desktop\Fetch> Receipts.py ...
51 # Check if the null percentage is greater than 0
52 if value > 0:
53     # Print the column name and its null percentage
54     print(key, ":", value * 100)
55 # unique_points_earned = receipts["pointsEarned"].unique()
56 # print(unique_points_earned)
57
58 # Check for duplicates based on all columns except "id/sold"
59 subset_columns = [col for col in receipts.columns if receipts[col].dtype != 'o']
60
61 # Check for duplicates based on the subset of columns
62 duplicateRowsDF = receipts[receipts.duplicated(subset=subset_columns, keep=False)]
63 print("Duplicate Rows except first occurrence based on selected columns are:")
64 print(duplicateRowsDF)
65
```

Python Debug Console

```
PS C:\Users\thaka\OneDrive\Desktop\Fetch> & 'c:\Python312\python.exe' 'c:\Users\thaka\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundle\libs\debugpy\adapter\...\debugpy\launcher' '5878'
4' '-' 'c:\Users\thaka\OneDrive\Desktop\Fetch\Receipts.py'
Duplicate Rows except first occurrence based on selected columns are:
   id bonusPointsEarnedReason ... rewardsReceiptStatus totalSpent      userId
2  {'id': '5ff1e1f10a720f052300057a'}      5.0 All-receipts receipt bonus ... REJECTED      10.00 5ff1e1f1cf6c399c274b0b
4  {'id': '5ff1e1d20a7214ada1000561'}      5.0 All-receipts receipt bonus ... FINISHED      1.00 5ff1e194b6a9d73a3a9f1052
5  {'id': '5ff1e1e40a7214ada1000566'}     750.0 Receipt number 1 completed, bonus point schedu... FINISHED      3.25 5ff1e1e4cf6c399c274ac3
6  {'id': '5ff1e1c0a720f052300056f'}      5.0 All-receipts receipt bonus ... FINISHED      2.23 5ff1e194b6a9d73a3a9f1052
10 {'id': '5ff1e1c50a720f052300056c'}    100.0 Receipt number 6 completed, bonus point schedu... FINISHED      1.00 5ff1e194b6a9d73a3a9f1052
...
1114 {'id': '603c0c10a720f051000030e'}     25.0 COMPLETE_NONPARTNER_RECEIPT ... REJECTED     34.96 5fc961c3b8cfca11a077dd33
1115 {'id': '603c0b710a720f051000042a'}     NaN COMPLETE_NONPARTNER_RECEIPT ... SUBMITTED     NaN 5fc961c3b8cfca11a077dd33
1116 {'id': '603c1520a720f0510000411'}     NaN COMPLETE_NONPARTNER_RECEIPT ... SUBMITTED     NaN 5fc961c3b8cfca11a077dd33
1117 {'id': '603c210a7217c72c0004095'}     25.0 COMPLETE_NONPARTNER_RECEIPT ... REJECTED     34.96 5fc961c3b8cfca11a077dd33
1118 {'id': '603c4fe0a7217c72c000389'}     NaN COMPLETE_NONPARTNER_RECEIPT ... SUBMITTED     NaN 5fc961c3b8cfca11a077dd33

[922 rows x 15 columns]
PS C:\Users\thaka\OneDrive\Desktop\Fetch>
```

Examining percentage of different category values for categorical variables

```
File Edit Selection View Go Run Terminal Help
C:\Users\thaka> OneDrive\Desktop\Fetch> Receipts.py ...
59 # subset_columns = [col for col in receipts.columns if receipts[col].dtype != 'o']
60
61 # Check for duplicates based on the subset of columns
62 duplicateRowsDF = receipts[receipts.duplicated(subset=subset_columns, keep=False)]
63 print("Duplicate Rows except first occurrence based on selected columns are:")
64 # print(duplicateRowsDF)
65 # Calculate the frequency of reasons and convert to percentages
66 freq_reasons = 100 * (receipts['bonusPointEarnedReason'].value_counts() / len(receipts))
67
68 # Print the percentages formatted to two decimal places
69 print(freq_reasons.map('{:.2f}%'.format))
70
```

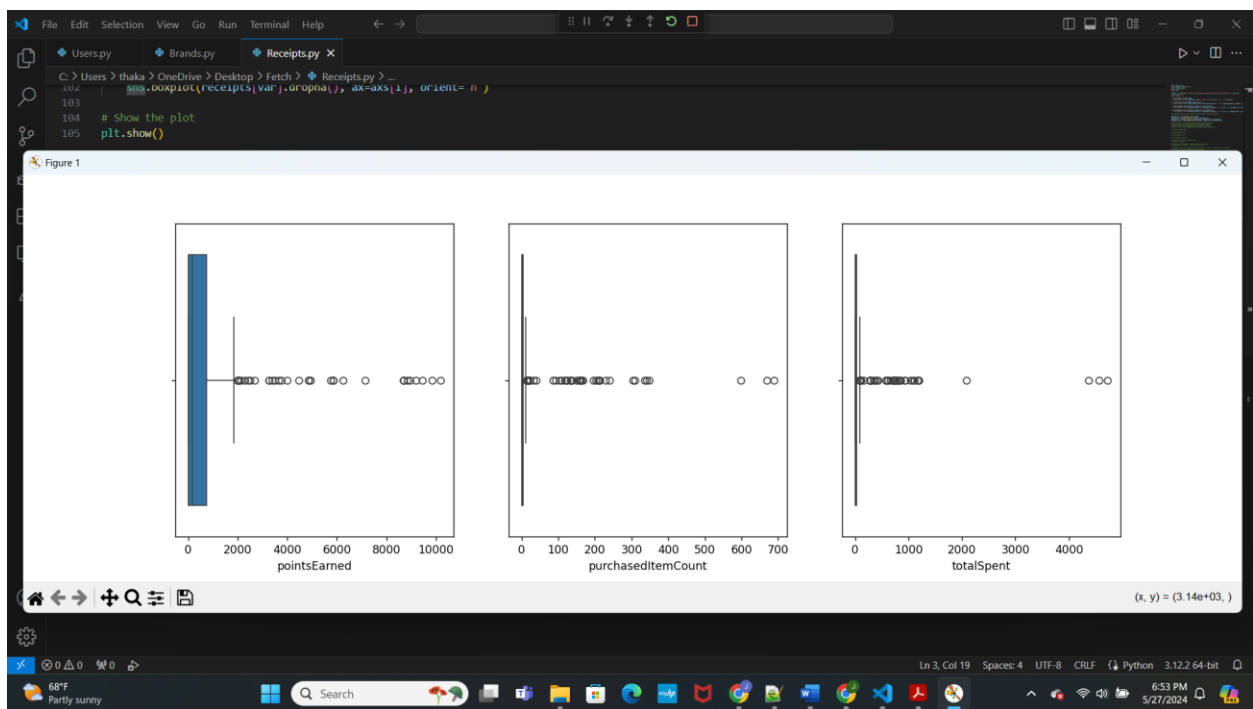
Python Debug Console

```
PS C:\Users\thaka\OneDrive\Desktop\Fetch> & 'c:\Python312\python.exe' 'c:\Users\thaka\.vscode\extensions\ms-python.debugpy-2024.6.0-win32-x64\bundle\libs\debugpy\adapter\...\debugpy\launcher' '5882'
4' '-' 'c:\Users\thaka\OneDrive\Desktop\Fetch\Receipts.py'
bonusPointsEarnedReason
All-receipts receipt bonus      16.35%
Receipt number 1 completed, bonus point schedule DEFAULT (5cfdcac3f693e0b50e83a36)  10.63%
COMPLETE_NONPARTNER_RECEIPT      6.34%
COMPLETE_NONPARTNER_RECEIPT      3.48%
Receipt number 3 completed, bonus point schedule DEFAULT (5cfdcac3f693e0b50e83a36)   2.77%
Receipt number 2 completed, bonus point schedule DEFAULT (5cfdcac3f693e0b50e83a36)   2.68%
Receipt number 5 completed, bonus point schedule DEFAULT (5cfdcac3f693e0b50e83a36)   2.41%
Receipt number 4 completed, bonus point schedule DEFAULT (5cfdcac3f693e0b50e83a36)   2.32%
Receipt number 6 completed, bonus point schedule DEFAULT (5cfdcac3f693e0b50e83a36)   1.61%
Name: count, dtype: object
PS C:\Users\thaka\OneDrive\Desktop\Fetch>
```

Plotting outliers using boxplot

```
File Edit Selection View Go Run Terminal Help
C:\Users> thaka > OneDrive > Desktop > Fetch > Receipts.py > ...
71 fig, axs = plt.subplots(ncols=3, nrows=1, figsize=(15, 5))
72
73 # Flatten the axs array for easier iteration
74 axs = axs.flatten()
75 # Define temporal variables
76 temporal = [
77     var for var in receipts.columns
78     if 'date' in var.lower() or 'time' in var.lower()
79 ]
80
81 # Define discrete variables
82 discrete = [
83     var for var in receipts.columns
84     if receipts[var].dtype != 'o' and len(receipts[var].unique()) < 20
85 ]
86
87 # Define categorical variables
88 categorical = [
89     var for var in receipts.columns
90     if receipts[var].dtype == 'o' and var not in temporal and var not in discrete
91 ]
92
93 # Define the list of continuous variables
94 continuous = [
95     var for var in receipts.columns
96     if var not in temporal + discrete + categorical
97 ]
98
99 # Iterate through each continuous variable
100 for i, var in enumerate(continuous):
101     # Filter out non-null values for the variable and create a boxplot
102     sns.boxplot(receipts[var].dropna(), ax=axs[i], orient='h')
103
104 # Show the plot
105 plt.show()
```

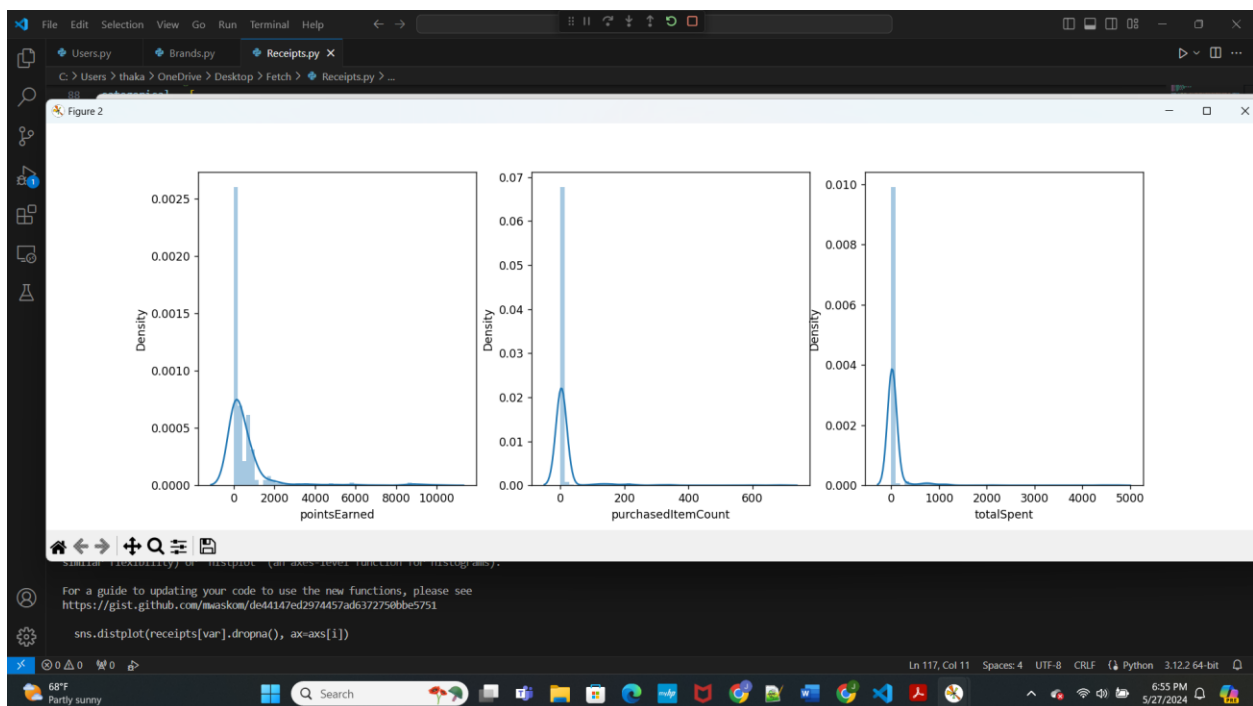
Ln 3, Col 19 Spaces: 4 UTF-8 CRLF Python 3.12.2 64-bit



Examining distributions of continuous variables

```
File Edit Selection View Go Run Terminal Help
C: > Users > thaka > OneDrive > Desktop > Fetch > Receipts.py > ...

104 # show the plot
105 # plt.show()
106 fig, axs = plt.subplots(ncols=3, nrows=1, figsize=(15, 5))
107
108 # Flatten the axs array for easier iteration
109 axs = axs.flatten()
110
111 # Iterate through each continuous variable
112 for i, var in enumerate(continuous):
113     # Filter out non-null values for the variable and create a distribution plot
114     sns.distplot(receipts[var].dropna(), ax=axs[i])
115
116 # Show the plot
117 plt.show()
```



Examining correlation between variables

```
File Edit Selection View Go Run Terminal Help
C:\Users\thaka> OneDrive\OneDrive\Fetch> Receipts.py ...
114 # sns.distplot(receipts[var].dropna(), ax=axes[i])
115
116 # show the plot
117 # plt.show()
118 # create subplots
119 # Drop non-numeric columns
120 numeric_columns = receipts.select_dtypes(include=np.number).columns
121 receipts_numeric = receipts[numeric_columns]
122
123 # create subplots
124 f, ax = plt.subplots(figsize=(10, 10))
125
126 # Calculate the correlation matrix
127 corr = receipts_numeric.corr()
128
129 # Create heatmap
130 sns.heatmap(
131     corr,
132     mask=np.zeros_like(corr, dtype=bool), # change np.bool to bool
133     square=True,
134     ax=ax
135 )
136
137 # Show the plot
138 plt.show()
```

