Αναλυτική Αναφορά
Liquor Sales Analysis(2016-2019)

Εισαγωγή

Η ακόλουθη αναλυτική αναφορά περιγράφει την μεθοδολογία, την ανάλυση και τα αποτελέσματα της επεξεργασίας του συνόλου δεδομένων πωλήσεων αλκοολούχων ποτών για την περίοδο 2016-2019. Στόχος της ανάλυσης είναι:

1. Η αναγνώριση του πιο δημοφιλούς προϊόντος σε κάθε ταχυδρομικό κώδικα (zip_code).

2. Ο υπολογισμός του ποσοστού συνολικών πωλήσεων ανά κατάστημα.

Για την υλοποίηση της ανάλυσης χρησιμοποιήθηκε η γλώσσα Python και οι βιβλιοθήκες Pandas, Matplotlib, Seaborn και Plotly.

Ανάλυση Δεδομένων

Φόρτωση και Καθαρισμός Δεδομένων

Έκανα εξαγωγή των δεδομένων από το διαθέσιμο αρχείο CSV και μια αρχική επισκόπηση για τον εντοπισμό των χαρακτηριστικών του dataset.

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# Φόρτωση των δεδομένων (περίοδος 2012-2020)
url = "https://storage.googleapis.com/courses_data/Assignment%20CSV/finance_liquor_sales.csv"
df = pd.read_csv(url)

# Προεπισκόπηση των αρχικών δεδομένων
print("Προεπισκόπηση αρχικών δεδομένων:")
print(df.head())
```

Μετέτρεψα την στήλη date σε τύπο datetime.

```python
# Μετατροπή της στήλης 'date' σε τύπο datetime
df['date'] = pd.to_datetime(df['date'], errors='coerce')
```

Έκανα φιλτράρισμα των δεδομένων για την επιλεγμένη περίοδο (2016-2019) που μας ενδιαφέρει.

```python
# Φιλτράρισμα των δεδομένων για την περίοδο 2016-2019
df_filtered = df[(df['date'].dt.year >= 2016) & (df['date'].dt.year <= 2019)]
print("\nΔεδομένα μετά το φιλτράρισμα για τα έτη 2016-2019:")
print(df_filtered.head())
```

Έλεγξα με το df_filtered.isnull().sum() πόσες τιμές λείπουν σε κάθε στήλη και με το dropna() διέγραψα όλες τις γραμμές που περιέχουν έστω και μία ελλειπή τιμή.

```
# Έλεγχος για ελλείπουσες τιμές σε όλες τις στήλες
print("\nΈλεγχος για ελλείπουσες τιμές σε όλες τις στήλες:")
print(df_filtered.isnull().sum())

# Αφαίρεση όλων των γραμμών που περιέχουν ελλείπουσες τιμές
df_filtered = df_filtered.dropna()

print("\nΔεδομένα μετά την αφαίρεση γραμμών με ελλείπουσες τιμές:")
print(df_filtered.info())
```

Μετέτρεψα τις αριθμητικές τιμές στον κατάλληλο τύπο δεδομένων. Το zip_code και το bottles_sold σε int και το sales_dollars σε float.

```
# Καθαρισμός της στήλης 'zip_code' και μετατροπή της σε ακέραιο αριθμό
df_filtered['zip_code'] = df_filtered['zip_code'].astype(int)

# Μετατροπή αριθμητικών στηλών σε σωστούς τύπους
df_filtered['bottles_sold'] = df_filtered['bottles_sold'].astype(int)
df_filtered['sale_dollars'] = df_filtered['sale_dollars'].astype(float)

# Τελική επισκόπηση των καθαρισμένων δεδομένων
print("\nΠληροφορίες για τα καθαρισμένα δεδομένα:")
print(df_filtered.info())
print("\nΠροεπισκόπηση των καθαρισμένων δεδομένων:")
print(df_filtered.head())
```

Εύρεση του πιο δημοφιλούς προϊόντος ανά zip code

Για την εύρεση του πιο δημοφιλούς προϊόντος σε κάθε περιοχή ομαδοποίησα τα δεδομένα ανά zip_code και item_number και υπολόγισα τις πωλήσεις (bottles_sold) ανά προϊόν και zip_code. Χρησιμοποίησα idxmax() για να πάρω το προϊόν με τις περισσότερες πωλήσεις.

```
# Task 1: Εύρεση του πιο δημοφιλούς προϊόντος ανά Zip Code

# Ομαδοποίηση κατά zip_code και item_number, υπολογισμός συνολικών πωλήσεων (bottles_sold)
popular_items = df_filtered.groupby(['zip_code', 'item_number'])['bottles_sold'].sum().reset_index()

# Εύρεση για κάθε zip_code του προϊόντος (item_number) με τις περισσότερες πωλήσεις (bottles_sold)
most_popular_per_zip = popular_items.loc[popular_items.groupby('zip_code')['bottles_sold'].idxmax()]

print("\nΠιο δημοφιλές προϊόν (με βάση το item_number) ανά Zip Code:")
print(most_popular_per_zip)
```

Οπτικοποίησα με scatter plot διάγραμμα που δείχνει τον αριθμό των πωλήσεων ανά προϊόν και zip_code.

```
# Visualization for Task 1

# Seaborn Scatter Plot - Bottles Sold Visualization
plt.figure(figsize=(10, 6))
sns.scatterplot(x=most_popular_per_zip.index,  # Άξονας X το index
                y=most_popular_per_zip['bottles_sold'],
                alpha=0.7, legend=None)

# Προσθήκη ετικετών για τις κουκίδες που έχουν bottles_sold > 200
for i in range(len(most_popular_per_zip)):
    if most_popular_per_zip['bottles_sold'].iloc[i] > 200:
        plt.annotate(most_popular_per_zip['item_number'].iloc[i],
                     xy=(most_popular_per_zip.index[i], most_popular_per_zip['bottles_sold'].iloc[i]),
                     textcoords="offset points", xytext=(0, 10), ha='center')

plt.title(label='Bottles Sold', fontsize=16)
plt.xlabel(xlabel='Index', fontsize=14)
plt.ylabel(ylabel='Bottles Sold', fontsize=14)
plt.tight_layout()
plt.show()
```

Υπολογισμός ποσοστού πωλήσεων ανά κατάστημα

Για κάθε κατάστημα, υπολόγισα το ποσοστό των συνολικών πωλήσεων.

```
# Task 2: Υπολογισμός ποσοστού συνολικών πωλήσεων ανά κατάστημα

# Υπολογισμός του συνολικού ποσού πωλήσεων για κάθε κατάστημα
sales_per_store = df_filtered.groupby(['store_number', 'store_name'])['sale_dollars'].sum().reset_index()

# Υπολογισμός του συνολικού ποσού πωλήσεων για όλα τα καταστήματα
total_sales = sales_per_store['sale_dollars'].sum()

# Υπολογισμός του ποσοστού πωλήσεων ανά κατάστημα
sales_per_store['sales_percentage'] = (sales_per_store['sale_dollars'] / total_sales) * 100

# Προεπισκόπηση των αποτελεσμάτων
print("\nΠοσοστά πωλήσεων ανά κατάστημα:")
print(sales_per_store)
```

Οπτικοποίησα με χρήση Plotly σε οριζόντιο bar chart που δείχνει ποιο κατάστημα έχει το μεγαλύτερο ποσοστό πωλήσεων.

```
# Visualization for Task 2

# Plotly Bar Chart - Sales Percentage Visualization
fig = px.bar(sales_per_store,
             x='sales_percentage',
             y='store_name',
             orientation='h',
             title='%Sales by Store',
             labels={'sales_percentage': 'Sales(%)', 'store_name': 'Store Name'},
             color='sales_percentage',  # Χρώμα ανάλογα με το sales_percentage
             color_continuous_scale='Viridis',  # Παλέτα χρωμάτων
             text='sales_percentage')  # Εμφάνιση ποσοστού στις μπάρες

fig.update_traces(texttemplate='%{text:.2f}', textposition='outside')
fig.update_layout(title_x=0.5)
fig.show()
```

Αποτελέσματα

Τα αρχικά δεδομένα περιλάμβαναν 24 στήλες. Παρακάτω παρουσιάζονται οι πρώτες 5 γραμμές των αρχικών δεδομένων:

```
Προεπισκόπηση αρχικών δεδομένων:
  invoice_and_item_number        date  ...  volume_sold_liters volume_sold_gallons
0         INV-31797900035  2020-11-10  ...                0.37                0.09
1         INV-23548800092  2019-11-27  ...                6.00                1.58
2         INV-23609300026  2019-12-02  ...                1.12                0.29
3         INV-39482900037  2021-08-24  ...                0.24                0.06
4         INV-39520400088  2021-08-25  ...                4.00                1.05

[5 rows x 24 columns]
```

Μετά το φιλτράρισμα , τα δεδομένα περιλαμβάνουν μόνο εγγραφές που αντιστοιχούν στα έτη 2016-2019:

```
Δεδομένα μετά το φιλτράρισμα για τα έτη 2016-2019:
  invoice_and_item_number       date  ...  volume_sold_liters volume_sold_gallons
1         INV-23548800092 2019-11-27  ...                6.00               1.58
2         INV-23609300026 2019-12-02  ...                1.12               0.29
6            S30390600011 2016-01-26  ...                1.20               0.32
7            S30348700047 2016-01-25  ...                4.00               1.06
8            S30466200002 2016-02-01  ...              120.00              31.70

[5 rows x 24 columns]
```

Εντοπίστηκαν στα δεδομένα κάποιες ελλείψεις στις στήλες country_number, category, category_name :

```
Ελεγχος για ελλείπουσες τιμές σε όλες τις στήλες:
invoice_and_item_number    0
date                       0
store_number               0
store_name                 0
address                    0
city                       0
zip_code                   0
store_location             9
county_number              1
county                     1
category                   3
category_name              9
vendor_number              0
vendor_name                0
item_number                0
item_description           0
pack                       0
bottle_volume_ml           0
state_bottle_cost          0
state_bottle_retail        0
bottles_sold               0
sale_dollars               0
volume_sold_liters         0
volume_sold_gallons        0
dtype: int64
```

Τα δεδομένα μετά τον καθαρισμό περιλαμβάνουν 60 εγγραφές χωρίς ελλείψεις.

```
Δεδομένα μετά την αφαίρεση γραμμών με ελλείπουσες τιμές:
<class 'pandas.core.frame.DataFrame'>
Index: 60 entries, 1 to 197
Data columns (total 24 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   invoice_and_item_number  60 non-null     object
 1   date                     60 non-null     datetime64[ns]
 2   store_number             60 non-null     int64
 3   store_name               60 non-null     object
 4   address                  60 non-null     object
 5   city                     60 non-null     object
 6   zip_code                 60 non-null     float64
 7   store_location           60 non-null     object
 8   county_number            60 non-null     float64
 9   county                   60 non-null     object
 10  category                 60 non-null     float64
 11  category_name            60 non-null     object
 12  vendor_number            60 non-null     float64
 13  vendor_name              60 non-null     object
 14  item_number              60 non-null     int64
 15  item_description         60 non-null     object
 16  pack                     60 non-null     int64
 17  bottle_volume_ml         60 non-null     int64
 18  state_bottle_cost        60 non-null     float64
 19  state_bottle_retail      60 non-null     float64
 20  bottles_sold             60 non-null     int64
 21  sale_dollars             60 non-null     float64
 22  volume_sold_liters       60 non-null     float64
```

```
 22  volume_sold_liters       60 non-null     float64
 23  volume_sold_gallons      60 non-null     float64
dtypes: datetime64[ns](1), float64(9), int64(5), object(9)
memory usage: 11.7+ KB
None
```

Πληροφορίες και προεπισκόπηση των καθαρισμένων δεδομένων.

```
Πληροφορίες για τα καθαρισμένα δεδομένα:
<class 'pandas.core.frame.DataFrame'>
Index: 60 entries, 1 to 197
Data columns (total 24 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   invoice_and_item_number  60 non-null     object
 1   date                    60 non-null     datetime64[ns]
 2   store_number            60 non-null     int64
 3   store_name              60 non-null     object
 4   address                 60 non-null     object
 5   city                    60 non-null     object
 6   zip_code                60 non-null     int64
 7   store_location          60 non-null     object
 8   county_number           60 non-null     float64
 9   county                  60 non-null     object
 10  category                60 non-null     float64
 11  category_name           60 non-null     object
 12  vendor_number           60 non-null     float64
 13  vendor_name             60 non-null     object
 14  item_number             60 non-null     int64
 15  item_description        60 non-null     object
 16  pack                    60 non-null     int64
 17  bottle_volume_ml        60 non-null     int64
 18  state_bottle_cost       60 non-null     float64
 19  state_bottle_retail     60 non-null     float64
 20  bottles_sold            60 non-null     int64
 21  sale_dollars            60 non-null     float64
 22  volume_sold_liters      60 non-null     float64
```

```
 23  volume_sold_gallons     60 non-null     float64
dtypes: datetime64[ns](1), float64(8), int64(6), object(9)
memory usage: 11.7+ KB
None
```

```
Προεπισκόπηση των καθαρισμένων δεδομένων:
   invoice_and_item_number       date  ...  volume_sold_liters volume_sold_gallons
1         INV-23548800092 2019-11-27  ...                 6.0               1.58
6            S30390600011 2016-01-26  ...                 1.2               0.32
7            S30348700047 2016-01-25  ...                 4.0               1.06
8            S30466200002 2016-02-01  ...               120.0              31.70
9         INV-16481100198 2018-12-20  ...                 6.0               1.58

[5 rows x 24 columns]
```

Βρέθηκε το προϊόν με τις περισσότερες πωλήσεις για κάθε περιοχή ανά ταχυδρομικό κώδικα.

```
Πιο δημοφιλές προϊόν (με βάση το item_number) ανά Zip Code:
    zip_code  item_number  bottles_sold
0      50010       946574           288
1      50022        86587             4
2      50131        38089            48
4      50158        67557             6
5      50263         3135            84
6      50265        67526            72
7      50266          250            90
8      50314        86251           240
9      50317        56193            24
12     50320       973627           120
13     50327        43040           102
15     50401       986845            48
16     50461        35918            30
17     50501        38176           108
18     50588        84617             4
19     50662        86739             8
20     50702        77487           768
22     50703          168           180
25     50707        15626            60
26     50801        43037             5
27     51106        67527           240
30     51247        86112             6
31     51401        27189            18
37     51501        86251            48
38     51555        45247             2
```

```
39     52001        67557             4
40     52003        43031             5
41     52136        35917             2
42     52172        67524             1
44     52240        86251            60
46     52241        65750            48
47     52314        75087          1560
48     52338        27357            90
50     52402        86390           216
51     52411        41846            36
52     52556        86251             6
53     52601        35913            48
54     52627        67586            36
55     52732        45248            24
56     52761        82847             4
57     52804        48690             2
```

Βρέθηκε το ποσοστό των συνολικών πωλήσεων σε κάθε κατάστημα.

```
Ποσοστά πωλήσεων ανά κατάστημα:
    store_number  ... sales_percentage
0           2178  ...         0.017217
1           2465  ...         0.073447
2           2512  ...         1.248543
3           2515  ...         1.591330
4           2538  ...         1.427997
5           2544  ...         0.133530
6           2562  ...         2.764359
7           2571  ...         3.521471
8           2576  ...         0.058333
9           2587  ...         0.113697
10          2591  ...         0.040727
11          2593  ...         1.303589
12          2601  ...         0.038500
13          2616  ...         0.498908
14          2619  ...         5.961126
15          2633  ...         7.289957
16          2636  ...         0.124727
17          2641  ...         0.031500
18          2647  ...         0.859089
19          2665  ...         2.684601
20          2670  ...         0.890907
21          2843  ...         0.074242
22          3162  ...         0.080818
23          3385  ...         1.221816
24          3447  ...        11.739191
25          3494  ...         9.190771
26          3524  ...         6.918533
```

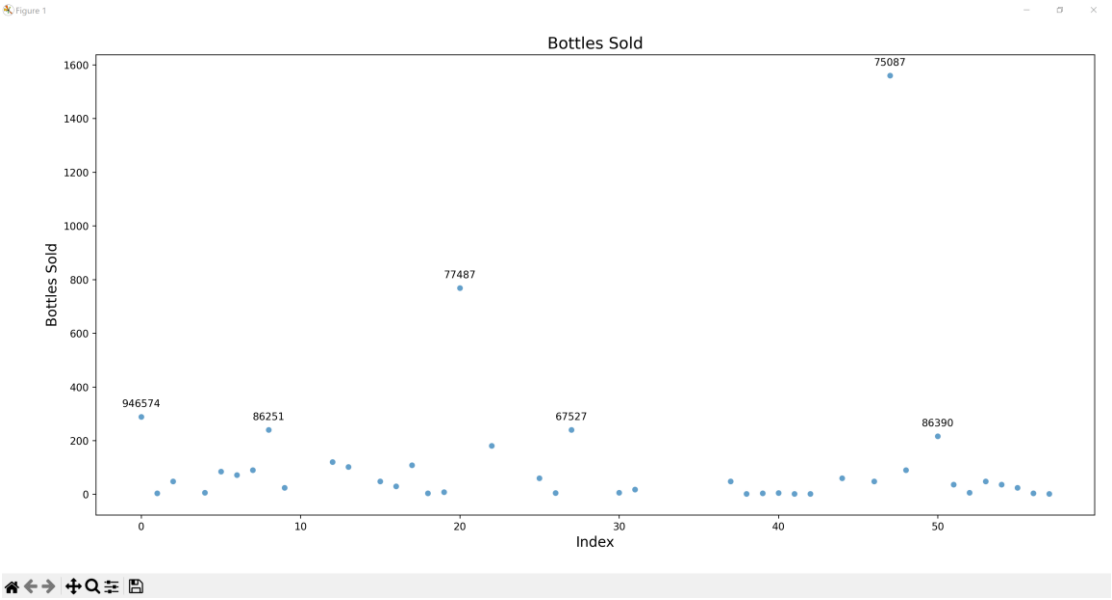
```
27          3549  ...         0.419363
28          3869  ...         0.011932
29          3963  ...         0.058757
30          4152  ...         0.023439
31          4153  ...         2.385299
32          4156  ...         0.185676
33          4158  ...         0.057485
34          4167  ...         0.089020
35          4291  ...         0.125947
36          4312  ...         0.307999
37          4559  ...         0.572726
38          4829  ...         1.539997
39          4867  ...         0.120520
40          4971  ...         4.056811
41          5003  ...         0.031783
42          5063  ...         0.112000
43          5102  ...        20.541781
44          5146  ...         2.385087
45          5204  ...         0.144242
46          5244  ...         0.132788
47          5416  ...         0.198810
48          5446  ...         0.037121
49          9001  ...         6.562488

[50 rows x 4 columns]

Process finished with exit code 0
```

Συμπεράσματα

Η ανάλυση των δεδομένων επέτρεψε την εξαγωγή χρήσιμων συμπερασμάτων σχετικά με τις πωλήσεις αλκοολούχων ποτών. Οι πιο δημοφιλείς κωδικοί προϊόντων και τα καταστήματα με τις υψηλότερες πωλήσεις εντοπίστηκαν.

Visualization Dashboard

- Για το Task 1



- Για το Task 2