


Instacart Market Basket Analysis

 쏘카캠퍼스 4기 **DATAthon**

3조 Pandas

김영준 노현정 김민지 이재상

Index



Data Overview



Subject




EDA & Visualization



Conclusion





Data Overview

 Featured Prediction Competition

Instacart Market Basket Analysis

Which products will an Instacart consumer purchase again?

 Instacart · 2,621 teams · 6 years ago



\$25,000
Prize Money

[Overview](#) [Data](#) [Code](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [Submissions](#) [Late Submission](#) [...](#)

Overview

Start


May 17, 2017

Close

Aug 15, 2017

Merger & Entry

Description

Whether you shop from meticulously planned grocery lists or let whimsy  guide your grazing, our unique food rituals define who we are. Instacart, a grocery ordering and delivery app, aims to make it easy to fill your refrigerator and pantry with your personal favorites and staples when you need them. After selecting products through the Instacart app, personal shoppers review your order and do the in-store shopping and delivery for you.

Instacart's data science team plays a big part in providing this delightful shopping experience. Currently they use transactional data to develop models that predict which products a user will buy again, try for the first time, or add to their cart next during a session. Recently, Instacart open sourced this data - see their blog post on [3 Million Instacart Orders, Open Sourced](#).

Competition Host

Instacart



Prizes & Awards

\$25,000

Awards Points & Medals

Participation

2,621 Competitors

2,621 Teams

39,863 Entries

Tags

Food

Table of Contents

Description

Evaluation

Prizes



Instacart Market Basket Analysis

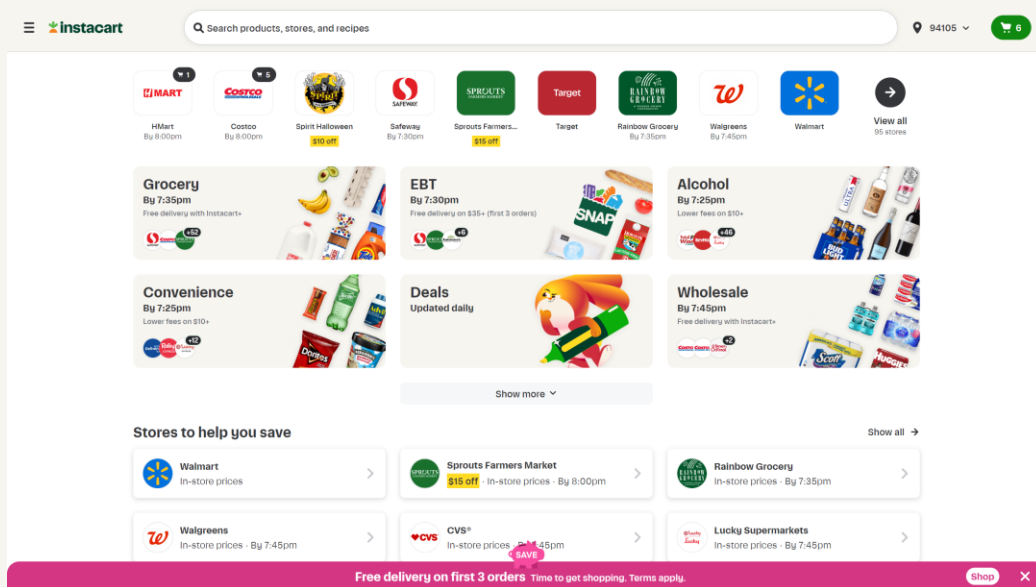
- Data From Kaggle
- Instacart의 소비자의 구매 정보, 상품정보, 상품 분류 정보, 진열 정보 등을 제공
- 제공 파일
 - aisles.csv
 - departments.csv
 - order_products__prior.csv
 - order_products__train.csv
 - orders.csv
 - products.csv
 - sample_submission.csv



Instacart Market Basket Analysis

- **Instacart**

- 온라인을 기반으로 하며, 식료품 배송 서비스를 제공하는 미국 기업
- 고객이 Instacart를 통해 주변 슈퍼마켓 및 식료품 매장의 식료품을 주문을 하면, Shopper가 대신 장을 봐준 뒤 배송



▶ 실제 Instacart 사이트



Instacart Market Basket Analysis

- **orders.csv**
 - 전체 주문 정보
- **order_products__prior.csv**
 - 상세 주문 내역(각 유저의 가장 최근 주문을 제외한 과거 주문에 관한)
 - 각 order에 따른 상세 구매 품목 목록과 해당 제품의 재구매율 등을 포함
- **order_products__train.csv**
 - 각 유저가 가장 최근에 주문한 1건에 대한 상세 주문 내역(test로 분류된 최근 주문 제외)
 - 이하 order_products__prior와 동일
- **products.csv**
 - 상품 정보
- **departments.csv**
 - 상품 분류 정보
- **aisles.csv**
 - 상품 2차 분류 정보



Subject

- 마트 인기 품목의 **구매왕** 찾기!
- **유기농** 제품을 파헤쳐 보자!
- 사고 또 사고! **재구매왕**
- 마트의 **Best 코너 & Worst 코너**
- 마트에 자주 방문하는 **단골** 찾기
- 판매 **키워드** 분석!



마트 인기 품목의 구매왕 찾기!

• 마트 인기 품목 분석

```
top_product = df['product_name'].value_counts().head(10)
top_product
```

Python

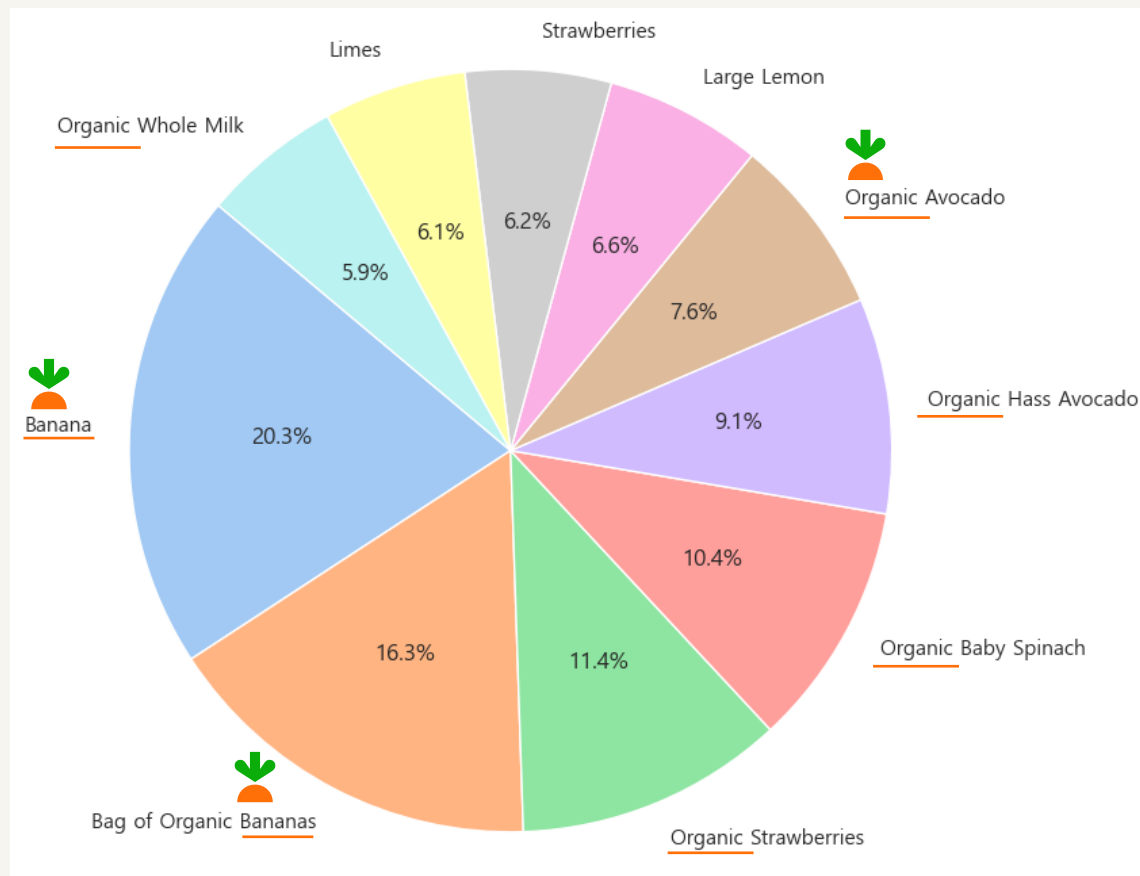
product_name	
Banana	491291
Bag of Organic Bananas	394930
Organic Strawberries	275577
Organic Baby Spinach	251705
Organic Hass Avocado	220877
Organic Avocado	184224
Large Lemon	160792
Strawberries	149445
Limes	146660
Organic Whole Milk	142813

Name: count, dtype: int64

```
labels = top_product.index
sizes = top_product.values

plt.figure(figsize=(8, 8))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.axis('equal')
plt.show()
```

Python





마트 인기 품목의 구매왕 찾기!

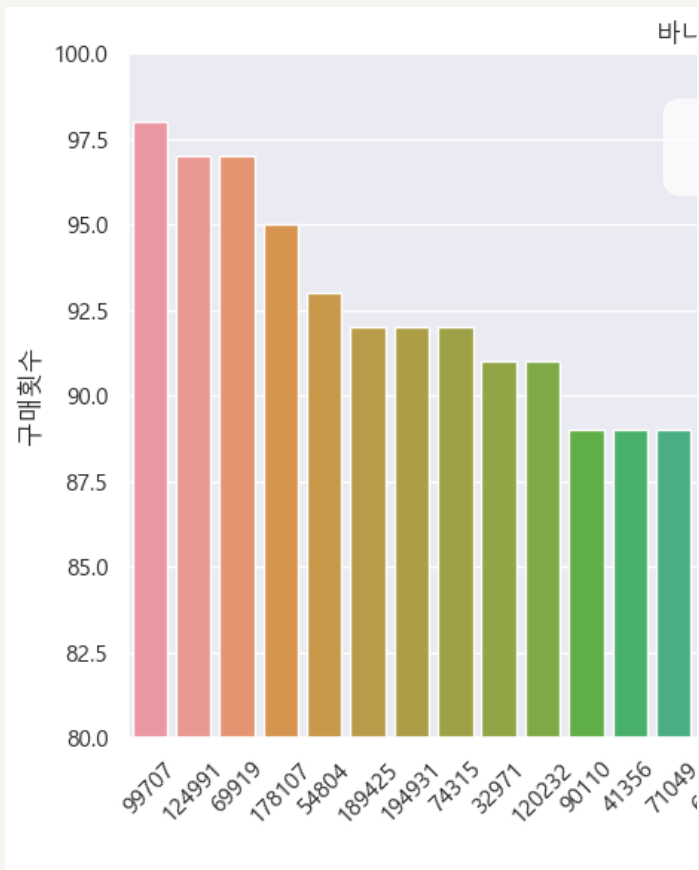
• 바나나 구매왕! 🐒 ★

```
# 전체 판매 품목중 Banana 단어가 들어간 품목 골라내기
banana_df =
product[product['product_name'].str.contains('Banana')].reset_index(drop=True)

# Banana가 들어간 품목중 순수 Banana만 골라냄
selected_products = ["Green Bananas", "Banana", "Bag of Organic Bananas", "Red Banana", "Baby Banana", "Baby Bananas", "Manzano Banana", "Organic Banana", "Bananas"]
banana_df2 = df[df['product_name'].isin(selected_products)]

# 순수 Banana 구매 고객중 상위 30명 선정
banana_df3=banana_df2['user_id'].value_counts().sort_values(ascending=False).head(30)

# 시각화
plt.figure(figsize=(10, 6))
sns.barplot(x=banana_df3.index.astype(str), y=banana_df3.values)
plt.title('바나나 구매 고객 탑30')
plt.xlabel('user_id')
plt.ylabel('구매횟수')
plt.ylim(80, 100)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



- User 99707
- User 124991



마트 인기 품목의 구매왕 찾기!

• 바나나 구매왕! 🐒 ⭐

```
# banana데이터에서 재주문횟수가 제일 높은 고객만 도출
banana_reorder = banana_df2.groupby('user_id')['reordered'].sum()
banana_reorder = banana_reorder.sort_values(ascending=False)
```

시각화

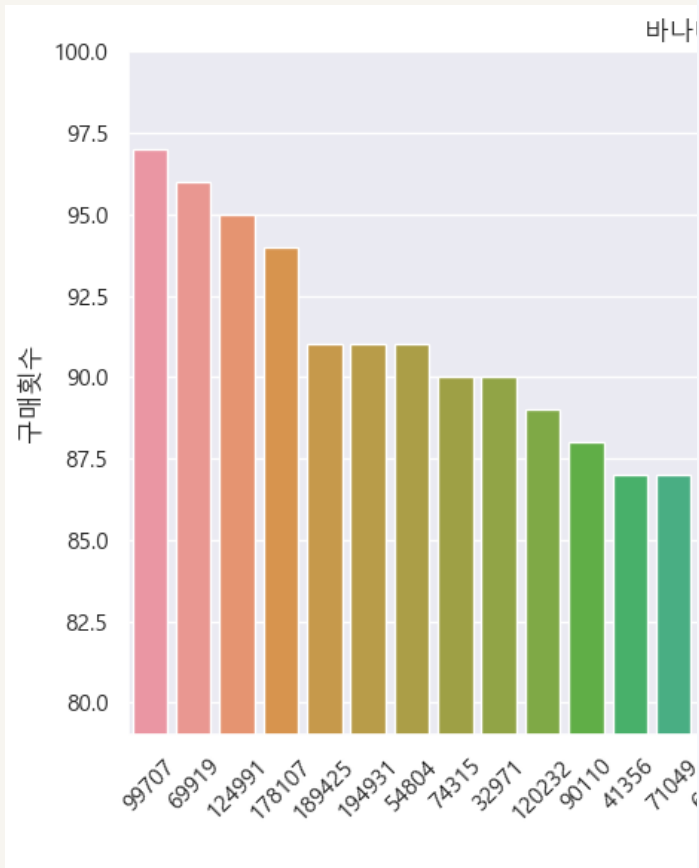
```
plt.figure(figsize=(10, 6))

sns.barplot(x=banana_reorder.index[:30].astype(str),
            y=banana_reorder.values[:30])
plt.title('바나나 재구매 고객 탑30')
plt.xlabel('user_id')
plt.ylabel('구매횟수')
plt.ylim(79, 100)
plt.xticks(rotation=45)
plt.tight_layout()

plt.show()
```



제품 수량 정보가 없어 재구매와 구매의 고객이 유사함





유기농 제품 파헤치기!

• 유기농 제품군! 🍓🥚

```
# 전체 판매 품목중 Organic단어가 포함된 품목선정
organic_df =
product[product['product_name'].str.contains('Organic')].reset_index(
drop=True)
organic_df
```

	product_id	product_name	aisle_id	department_id
0	23	Organic Turkey Burgers	49	12
1	33	Organic Spaghetti Style Pasta	131	9
2	41	Organic Sourdough Einkorn Crackers Rosemary	78	19
3	43	Organic Clementines	123	4
4	47	Onion Flavor Organic Roasted Seaweed Snack	66	6
...
5030	49606	Organic Natural Red	28	5
5031	49608	Certified Organic Spanish Style Rice With Quin...	4	9
5032	49638	Organic Superfirm Vacuum Packed Tofu	14	20
5033	49653	Organic Aromatherapeutic Moroccan Argan Oil Set	25	11
5034	49659	Organic Creamed Coconut	17	13

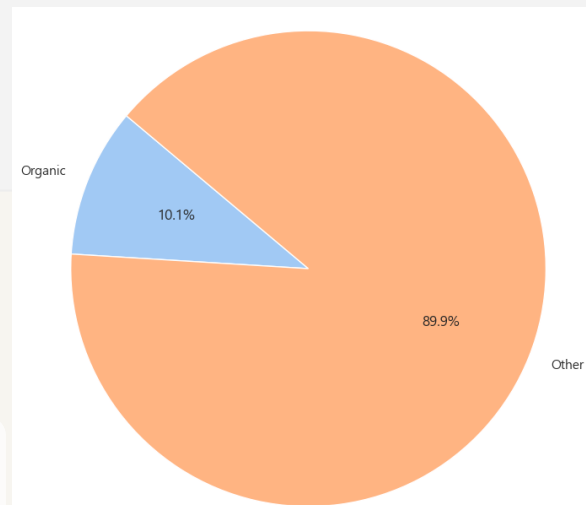
5035 rows × 4 columns

```
# 전체 품목에서의 유기농 품목의 비율
data = [len(organic_df), len(product) - len(organic_df)]
labels = ['Organic', 'Other']

plt.figure(figsize=(8, 8))
plt.pie(data, labels=labels, autopct='%1.1f%%', startangle=140)
```

```
plt.title('전체 품목중 유기농제품의 비율')
plt.axis('equal')
```

```
plt.show()
```



약 5만개의 판매 품목 중
유기농 제품은 10%



유기농 제품 파헤치기!

• 유기농 제품군 판매 비율 🍓 🍌

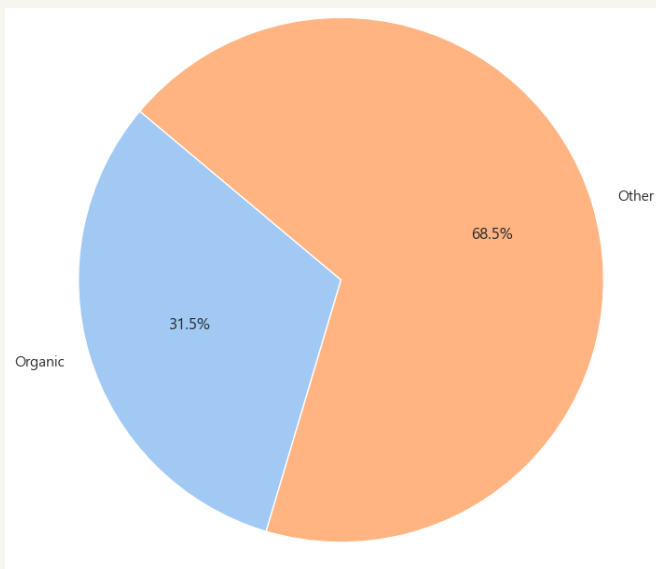
```
# 전체 판매내역중 유기농 제품 판매내역 확인
organic_df2 =
df[df['product_name'].isin(organic_df['product_name'].values)]

# 유기농제품이 들어있는 카테고리 확인
organic_df2['department'].unique()
```

```
array(['produce', 'snacks', 'pantry', 'beverages', 'deli',
      'dairy eggs', 'frozen', 'canned goods', 'bakery',
      'dry goods pasta', 'meat seafood', 'missing',
      'breakfast',
      'babies', 'personal care', 'international', 'bulk',
      'pets', 'other', 'household', 'alcohol'], dtype=object)
```



유기농 제품군이 모든 카테고리 내 포함되어 있음



```
# 전체 판매 내역중 유기농 판매 비율
print(len(organic_df2) / len(df))

data = [len(organic_df2), len(df) - len(organic_df2)]
labels = ['Organic', 'Other']

plt.figure(figsize=(8, 8))
plt.pie(data, labels=labels, autopct='%1.1f%%', startangle=140)

plt.title('유기농 vs 나머지 비율')
plt.axis('equal')

plt.show()
```



유기농 제품군이
전체 판매량 중
30% 이상 큰 비중 차지.



유기농 제품 파헤치기!

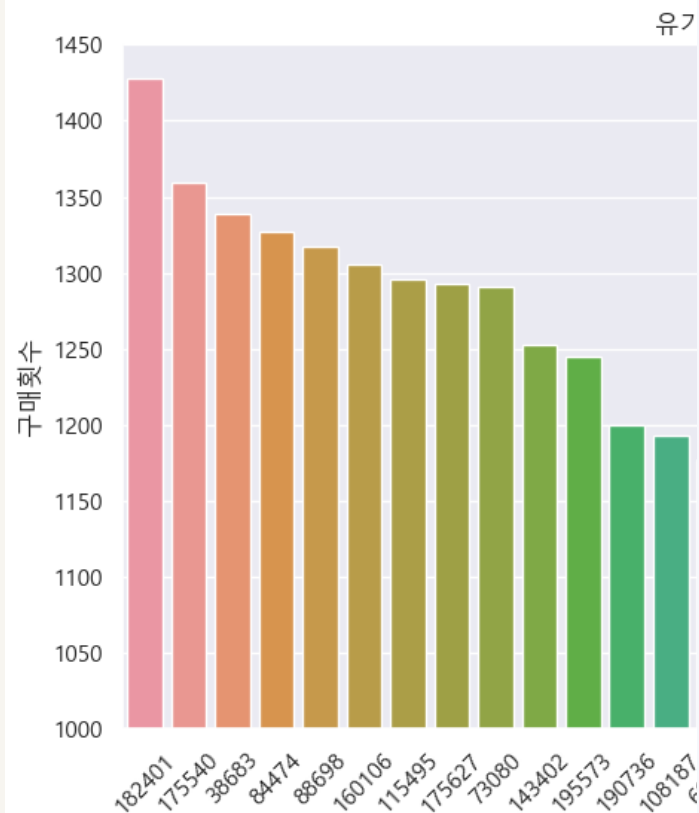
• 유기농 제품군 구매왕! 🍓 👁

```
# 유기농제품을 가장 많이 구매한 user_id 추출
organic_df3 =
organic_df2['user_id'].value_counts().sort_values(ascending=False).head(30)
```

```
# 시각화
plt.figure(figsize=(10, 6))
```

```
sns.barplot(x=organic_df3.index.astype(str), y=organic_df3.values)
plt.title('유기농 구매 고객 탑30')
plt.xlabel('user_id')
plt.ylabel('구매횟수')
plt.ylim(1000, 1450)
plt.xticks(rotation=45)
plt.tight_layout()
```

```
plt.show()
```



- User 182401
- User 175540



사고 또 사고! 재구매왕

• 재구매 비율이 높은 상품 선정! 📄 📊

```
# 재구매 비율이 높은 상품 중 판매비율이 유의미한 제품 판별(10,000이상 판매)
product_reorder_counts =
df.groupby('product_name')['reordered'].agg(['sum', 'count'])
product_reorder_counts.columns = ['reordered_count', 'total_count']

product_reorder_counts['reorder_ratio'] =
product_reorder_counts['reordered_count'] /
product_reorder_counts['total_count']

top_reorder_ratio =
product_reorder_counts.sort_values(by='reorder_ratio',
ascending=False)
filtered_top_reorder_ratio =
top_reorder_ratio[top_reorder_ratio['total_count'] >= 10000]
# 총 판매 갯수 10000개 이상만 보기
filtered_top_reorder_ratio.head(20)
```



재구매 비율이 높은 제품은
유제품(dairy eggs) '우유' 와 '요거트'
유제품 조차도 Organic 제품 선호함

	reordered_count	total_count	reorder_ratio
product_name			
Milk, Organic, Vitamin D	17753	20770	0.854742
Organic Reduced Fat Milk	31394	36869	0.851501
Banana	415166	491291	0.845051
Organic Lowfat 1% Milk	12914	15352	0.841193
Organic Whole Milk	8494	10102	0.840824
Organic Milk Reduced Fat, 2% Milkfat	10984	13119	0.837259
Bag of Organic Bananas	329275	394930	0.833755
Organic Fat Free Milk	22824	27402	0.832932
Organic Whole Milk	118684	142813	0.831045
0% Greek Strained Yogurt	11287	13651	0.826826
Organic Whole Milk with DHA Omega-3	11539	14051	0.821223
Extra Fancy Unsalted Mixed Nuts	8209	10030	0.818445
1% Lowfat Milk	11970	14692	0.814729
Italian Sparkling Mineral Water	26119	32069	0.814463
Pure Sparkling Water	24410	30009	0.813423



사고 또 사고! 재구매왕

• 카테고리 별 재구매율! 🍷 🍺

```
# 카테고리별 재주문 비율 확인하기
dpt_reorder_counts =
df.groupby('department')['reordered'].agg(['sum', 'count'])
dpt_reorder_counts.columns = ['reordered_count', 'total_count']

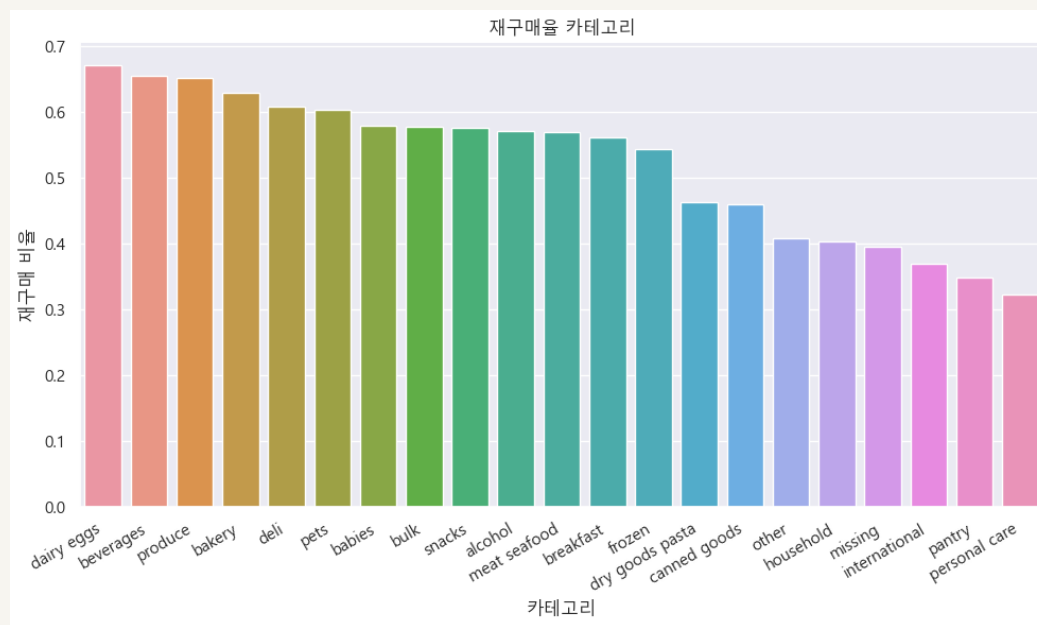
dpt_reorder_counts['reorder_ratio'] =
dpt_reorder_counts['reordered_count'] /
dpt_reorder_counts['total_count']

top_reorder_ratio =
dpt_reorder_counts.sort_values(by='reorder_ratio', ascending=False)
filtered_top_reorder_ratio =
top_reorder_ratio[top_reorder_ratio['total_count'] >= 1000]

# 시각화
plt.figure(figsize=(10, 6))

ax0 = sns.barplot(data=filtered_top_reorder_ratio,
x=filtered_top_reorder_ratio.index, y='reorder_ratio')
plt.title('재구매율 카테고리')
plt.xlabel('카테고리')
plt.ylabel('재구매 비율')
ax0.set_xticklabels(ax0.get_xticklabels(), rotation=30,
horizontalalignment='right')
plt.tight_layout()

plt.show()
```



유제품(dairy eggs), 음료(beverages) 재구매율이 높음
전체 판매량은 낮지만 재주문율은 높은 카테고리 :
pets(반려동물용품)



사고 또 사고! 재구매왕

• 사고 또 사는 재구매왕은? 🍷 🍷

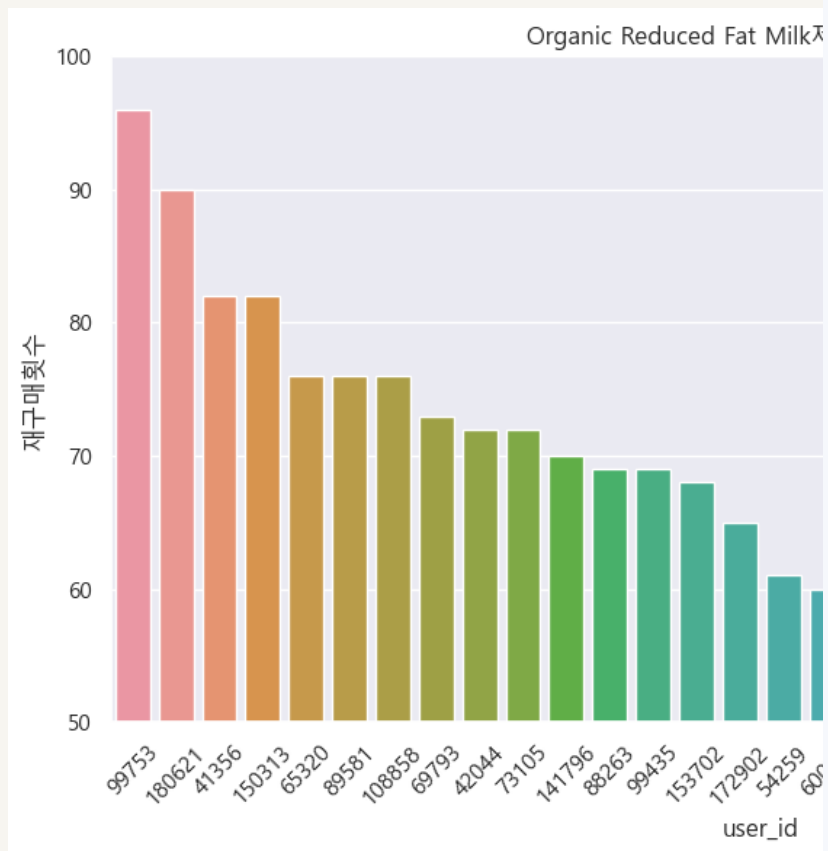
```
# 재주문율이 가장 높은 제품을 가장 많이 구매한 고객 추출
h_h = df[df['product_name'] == 'Organic Reduced Fat Milk']

h_h['user_id'].value_counts()

# 시각화
plt.figure(figsize=(10, 6))

sns.barplot(x=organic_df3.index.astype(str),
            y=organic_df3.values)
plt.title('유기농 구매 고객 탑30')
plt.xlabel('user_id')
plt.ylabel('구매횟수')
plt.ylim(1000, 1450)
plt.xticks(rotation=45)
plt.tight_layout()

plt.show()
```



- User 99753
- User 180621



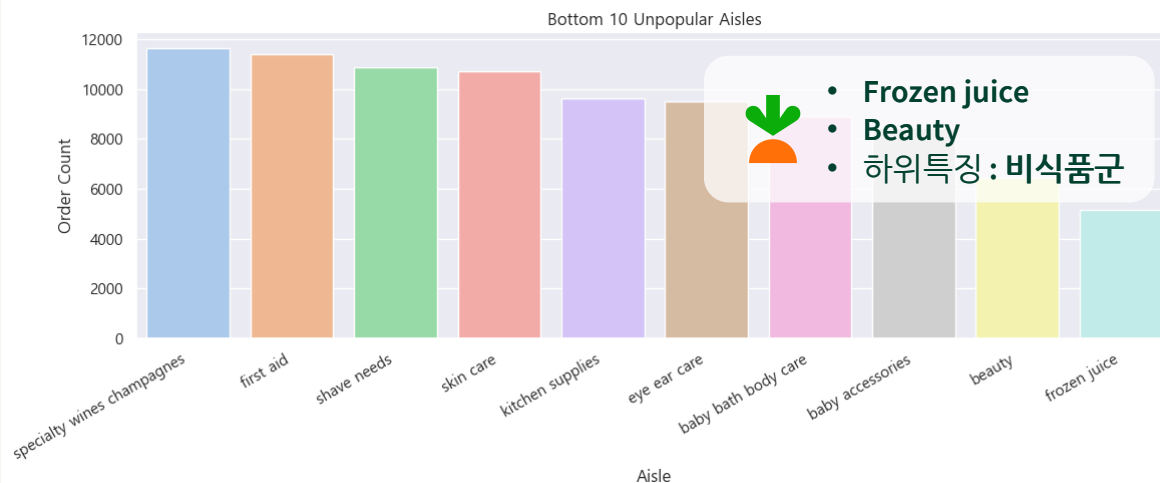
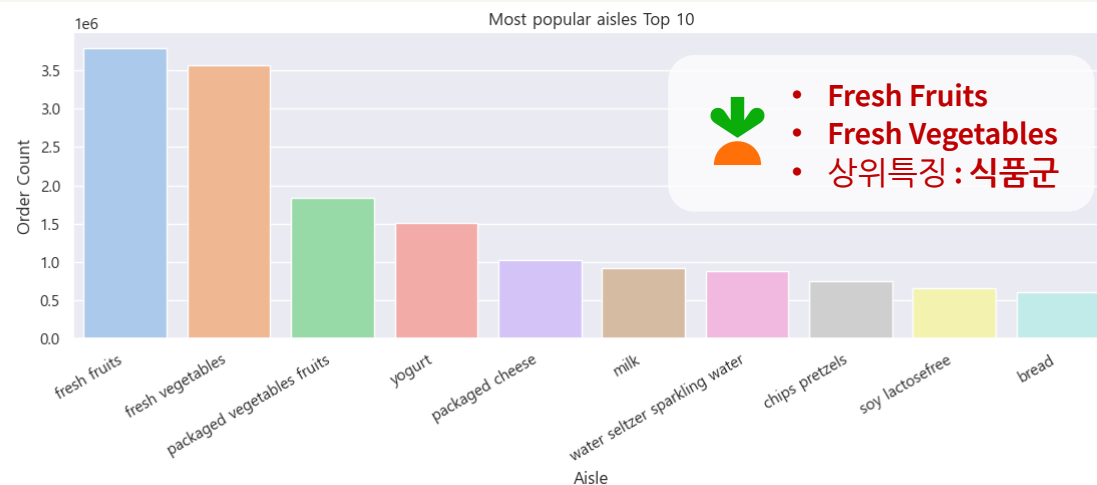
마트의 Best 코너 & Worst 코너

• 판매가 가장 🌶️ Hot한 Best & ❄️ Cool한 Worst 코너

```
# 제품 위치한 구역의 통로이름 별 주문 수량 top10 / bottom10 시각화
# 데이터프레임에서 상위 10개와 하위 10개 aisle 추출
top_10_aisles = aisle_counts.head(10)
bottom_10_aisles = aisle_counts.tail(10)
plt.figure(figsize=(12, 10))
```

```
# 가장 인기 있는 aisle 시각화
plt.subplot(2, 1, 1)
ax1 = sns.barplot(x=top_10_aisles["aisle"],
y=top_10_aisles["order_count"], data=top_10_aisles)
plt.title("Most popular aisles Top 10")
plt.xlabel("Aisle")
plt.ylabel("Order Count")
ax1.set_xticklabels(ax1.get_xticklabels(), rotation=30,
horizontalalignment='right')
```

```
# 가장 인기가 없는 aisle 시각화
plt.subplot(2, 1, 2)
ax2 = sns.barplot(x=bottom_10_aisles["aisle"],
y=bottom_10_aisles["order_count"], data=bottom_10_aisles)
plt.title("Bottom 10 Unpopular Aisles")
plt.xlabel("Aisle")
plt.ylabel("Order Count")
ax2.set_xticklabels(ax2.get_xticklabels(), rotation=30,
horizontalalignment='right')
plt.tight_layout()
plt.show()
```





마트의 Best 코너 & Worst 코너

• Worst 코너 ❄️ Frezen Juice와 Organic

```
# 냉동음료 제품중 Organic키워드가 들어간 제품 찾기
frozen_j = list(set(df[df['aisle'] == 'frozen juice']['product_name']))
o_f = []
for i in frozen_j:
    if "Organic" in i:
        o_f.append(i)

# 냉동 음료의 품목별 판매 개수 카운팅
Frozen_j_df = df[df['aisle'] == 'frozen juice']
F_j_d_c = frozen_j_df['product_name'].value_counts()

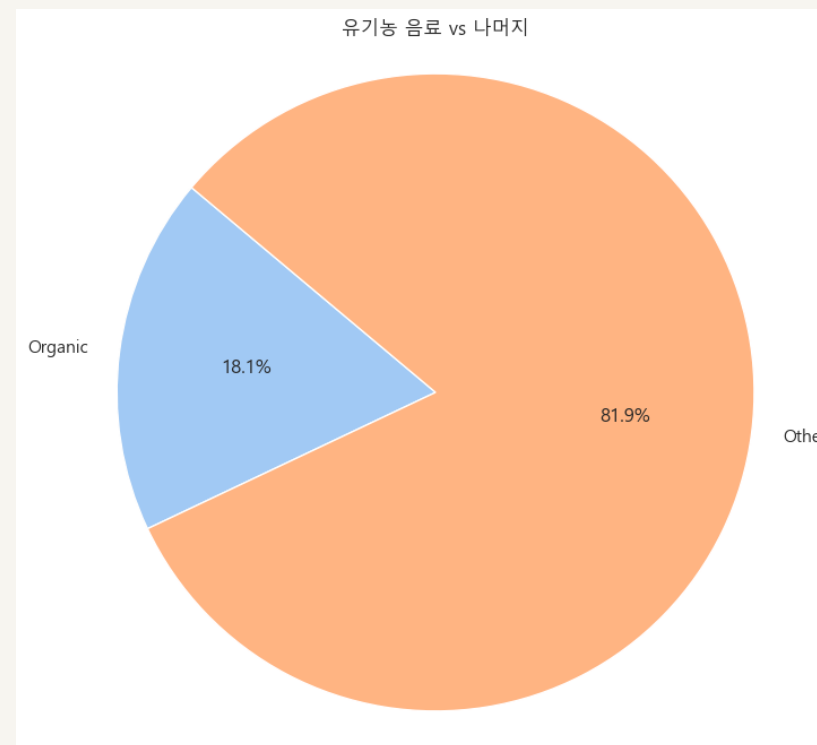
# 냉동 음료 중 Organic키워드가 들어간 제품의 판매 갯수만 출력
organic_fj = f_j_d_c[f_j_d_c.index.str.contains('Organic')].reset_index(drop=True)

# 시각화
data = [organic_fj.values.sum(), f_j_d_c.values.sum() - organic_fj.values.sum()]
labels = ['Organic', 'Other']

plt.figure(figsize=(8, 8))
plt.pie(data, labels=labels, autopct='%1.1f%%', startangle=140)

plt.title('유기농 음료 vs 나머지')
plt.axis('equal')

plt.show()
```



- 총 판매품목 47개 중 4개 만이 유기농 제품
- 하지만 판매 개수 비율로만 따지면 20%에 육박함



마트에 자주 방문하는 단골 고객 찾기!

• 단골 고객 선정하기!

```
# 재방문 평균 계산
revisit_mean =
df.groupby('user_id')['days_since_prior_order'].mean().reset_index()
revisit_mean.rename(columns={'days_since_prior_order':
'revisit_mean'}, inplace=True)

# 재방문 횟수 계산
revisit_cnt = df.groupby(['user_id',
'order_id'])['days_since_prior_order'].nunique().reset_index()
revisit_cnt =
revisit_cnt.groupby('user_id')['days_since_prior_order'].sum().reset_
index()
revisit_cnt.rename(columns={'days_since_prior_order':
'revisit_count'}, inplace=True)

# 데이터 병합
revisit_merge = pd.merge(revisit_mean, revisit_cnt, on='user_id')
revisit_merge['revisit_mean'] =
revisit_merge['revisit_mean'].astype(int)
revisit_merge
```



단골 고객 선정 기준

- 방문 주기가 짧고 방문횟수가 많은 회원
- 재방문까지 걸린 일자는 0일에서 30일까지로 분포
- 0일은 당일에 재방문한 경우
- 재방문까지 걸린 일자에 결측치는 첫 방문을 의미

	user_id	revisit_mean	revisit_count
0	1	19	10
1	2	18	14
2	3	11	11
3	4	15	4
4	5	12	4



마트에 자주 방문하는 단골 고객 찾기!

- 재방문 주기 평균 및 재방문 횟수 시각화

```
# 산포도 그리기
plt.figure(figsize=(10, 6))
plt.scatter(revisit_merge['revisit_mean'],
            revisit_merge['revisit_count'], alpha=0.5)

plt.xlabel('재방문 주기 평균')
plt.ylabel('재방문 횟수')
plt.title('재방문 주기 평균 & 재방문 횟수')
plt.grid(True)

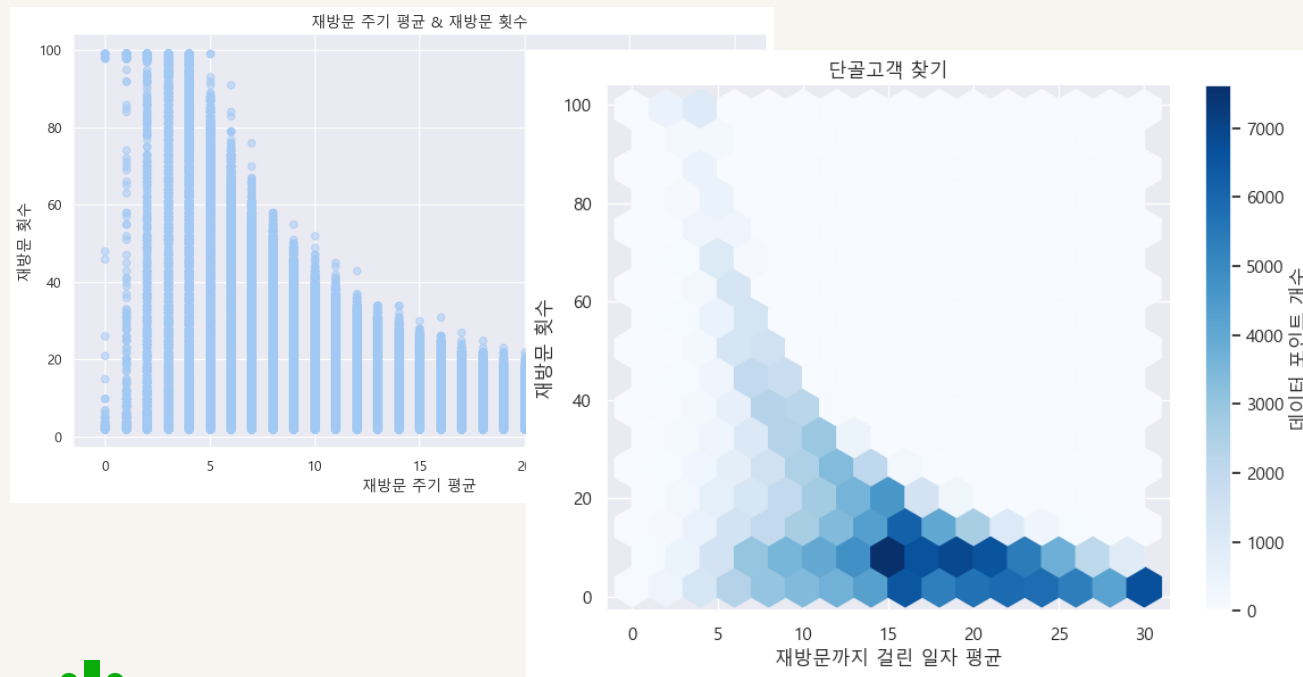
plt.show()

# Hexbin Plot 그리기
plt.figure(figsize=(8, 6))

x = revisit_merge['revisit_mean']
y = revisit_merge['revisit_count']
hb = plt.hexbin(x, y, gridsize=15, cmap='Blues')
plt.xlabel('재방문까지 걸린 일자 평균')
plt.ylabel('재방문 횟수')
plt.title('단골고객 찾기')

# 컬러 바 추가
cb = plt.colorbar(hb, label='데이터 포인트 개수')

plt.show()
```



- 유의미한 결과를 도출할 수 없음.
- 구매물품 횟수까지 포함해서 샘플링을 다시 시도



마트에 자주 방문하는 단골 고객 찾기!

- 구매 물품 기준 포함한 재방문 상위 고객

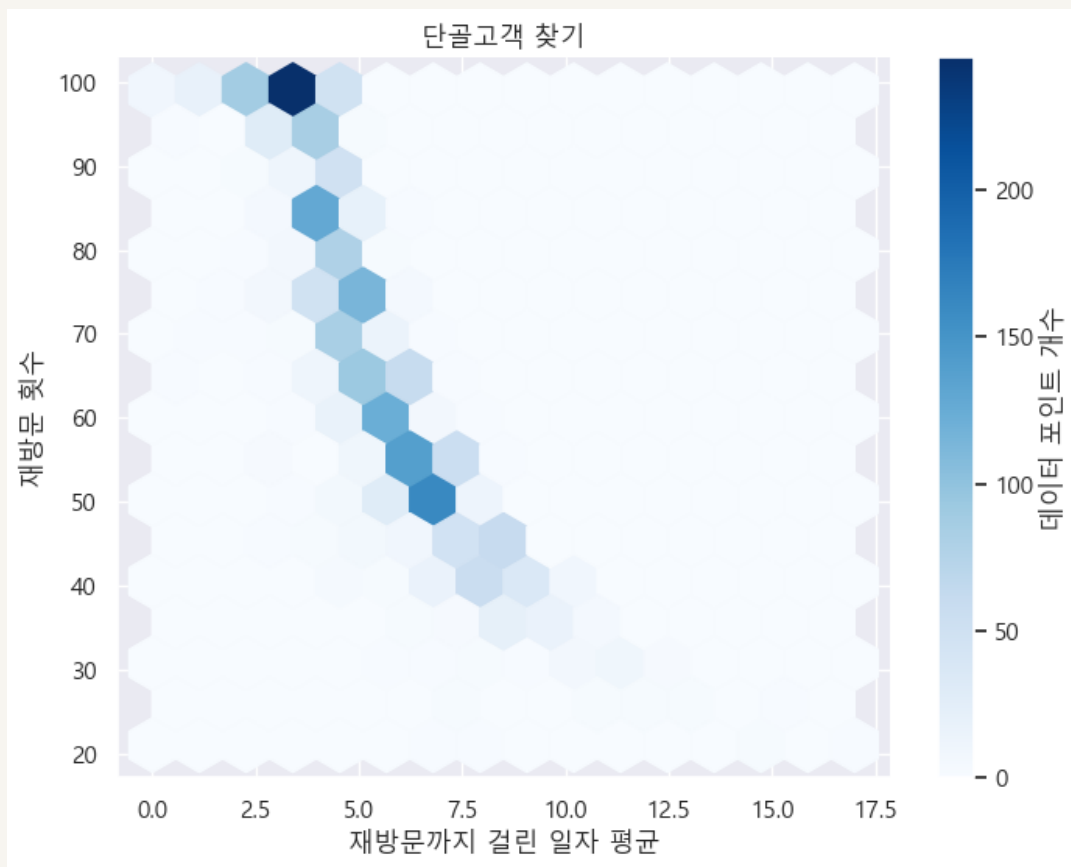
```
# 구매물품기준 상위 고객 출력
top_user = df['user_id'].value_counts()
top_user = top_user[top_user > 1000]
top_user = top_user.reset_index()
top_user.columns = ['user_id', 'purchase_count']

# 위의 표를 토대로 재방문 기준 출력
top_revisit = pd.merge(top_user, revisit_merge, on='user_id',
                        how='inner')

# Hexbin Plot 그리기
plt.figure(figsize=(8, 6))
x = top_revisit['revisit_mean']
y = top_revisit['revisit_count']
hb = plt.hexbin(x, y, gridsize=15, cmap='Blues')
plt.xlabel('재방문까지 걸린 일자 평균')
plt.ylabel('재방문 횟수')
plt.title('단골고객 찾기')
cb = plt.colorbar(hb, label='데이터 포인트 개수')
plt.show()
```



이상치에 해당하는 부분만 모아서 시각화 진행하여, 재방문 평균 일자가 비교적 짧으면서 재방문 횟수가 높은 고객 시각화





마트에 자주 방문하는 단골 고객 찾기!

- 단골 고객 찾기 total_score

```
# 재방문까지 걸린 일자 평균, 재방문 횟수, 구매물품 횟수 모두 고려하여  
total_score를 생성하여 단골 고객을 찾는다.
```

```
plt.figure(figsize=(8, 6))
```

```
# 첫 번째 Box Plot
```

```
plt.subplot(2, 1, 1)  
sns.boxplot(x='revisit_mean', data=top_revisit, width=0.4)  
plt.xlabel('재방문 주기 평균')  
plt.title('재방문 주기 평균에 대한 Box Plot')
```

```
# 두 번째 Box Plot
```

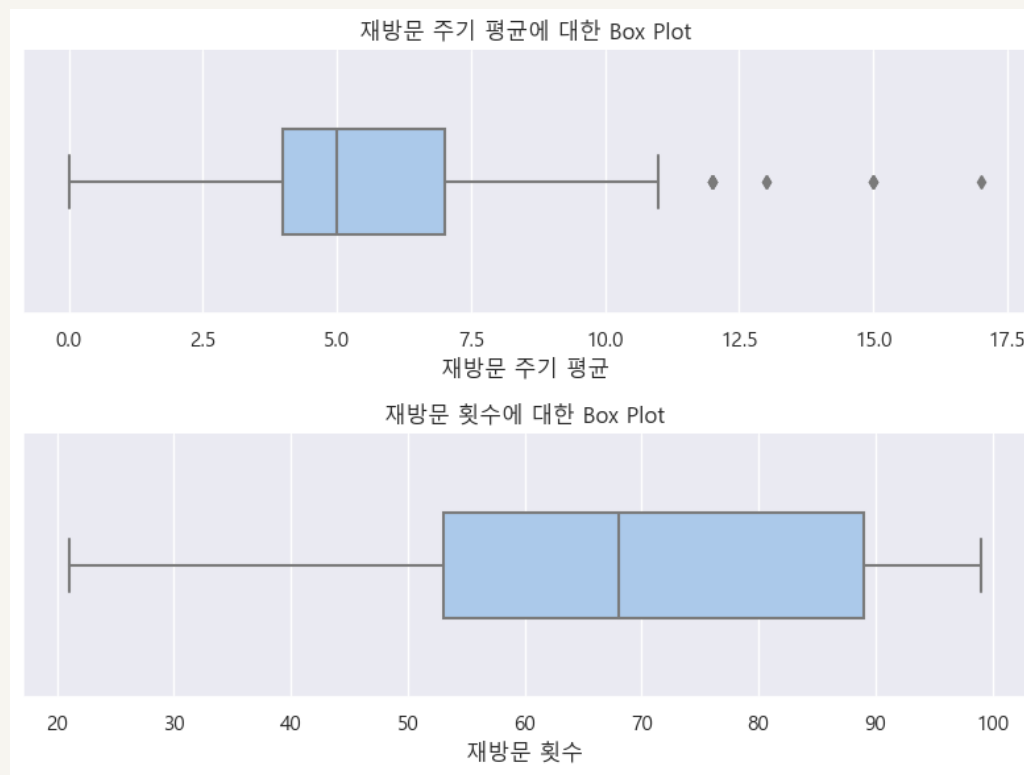
```
plt.subplot(2, 1, 2)  
sns.boxplot(x='revisit_count', data=top_revisit, width=0.4)  
plt.xlabel('재방문 횟수')  
plt.title('재방문 횟수에 대한 Box Plot')
```

```
# 그래프 표시
```

```
plt.tight_layout()  
plt.show()
```



재방문 주기 평균이 낮으면서 재방문 횟수가 높은 고객에 집중!





마트에 자주 방문하는 단골 고객 찾기!

- 단골 고객 찾기 total_score

```
# 재방문 주기 평균 10일 이하인 고객만 출력
revisit_10_under = top_revisit[top_revisit['revisit_mean'] < 10]

w1 = 1.5    # 재방문까지 걸린 일자 평균의 가중치
w2 = 1      # 재방문 횟수의 가중치
w3 = 1.3    # 상품 구매 횟수의 가중치

revisit_10_under['total_score'] = (revisit_10_under['revisit_mean'] * w1) + \
                                   (revisit_10_under['revisit_count'] * w2) + \
                                   (revisit_10_under['purchase_count'] * w3)

# 현재 total_score 값 중 최소값과 최대값 계산
min_total_score = revisit_10_under['total_score'].min()
max_total_score = revisit_10_under['total_score'].max()

# 설정 범위 (0~100)로 최소값과 최대값 설정
min_target = 0
max_target = 100

# total_score 값을 0에서 100 사이로 스케일링
revisit_10_under['scaled_total_score'] = ((revisit_10_under['total_score'] - min_total_score) /
                                           (max_total_score - min_total_score)) * (max_target - min_target) + min_target
revisit_10_under['scaled_total_score'] = revisit_10_under['scaled_total_score'].round(1)
```



- 가중치 설정
- MinMax 스케일링



마트에 자주 방문하는 단골 고객 찾기!

- 단골 고객 찾기 total_score

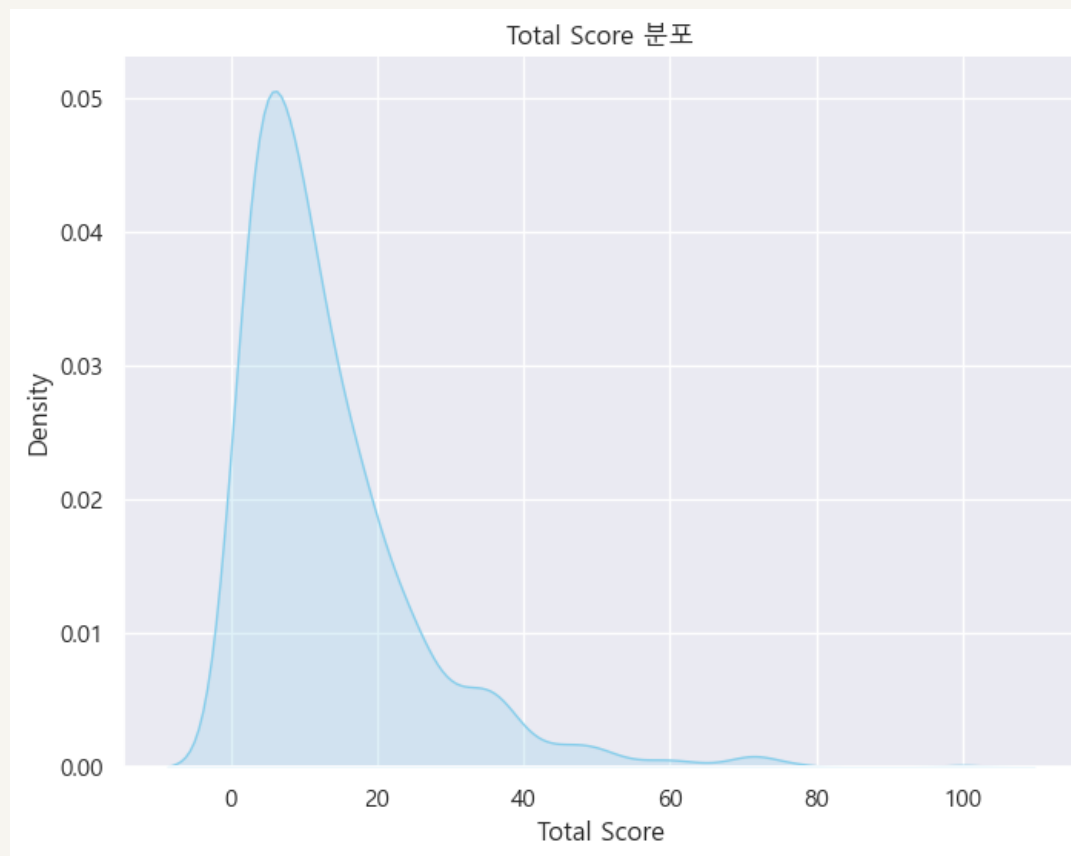
```
# 재방문 횟수에 대한 boxplot 결과를 참고하여 횟수가 80 이상인 고객으로 필터링
top_customers =
revisit_10_under[revisit_10_under['revisit_count'] >= 80]

# KDE 그래프 그리기
plt.figure(figsize=(8, 6))
sns.kdeplot(data=top_customers['scaled_total_score'], shade=True,
color='skyblue')
plt.xlabel('Total Score')
plt.ylabel('Density')
plt.title('Total Score 분포')

plt.show()
```



- total_score의 스케일링 하여 0~100점으로 상대평가
- 30점을 기준으로 빈도수가 많이 줄어듦
- 단골 고객 선정 기준을 total_score점수 30점 이상으로 설정





마트에 자주 방문하는 단골 고객 찾기!

• 최종 단골 고객 선정

```
# 단골 고객 선정
vip = top_customers[top_customers['scaled_total_score'] >= 30].reset_index(drop = True)
```

	user_id	purchase_count	revisit_mean	revisit_count	total_score	scaled_total_score
0	201268	3725	3	98	4945.0	100.0
1	164055	3089	3	99	4119.2	77.0
2	176478	2952	2	99	3939.6	72.0
3	186704	2936	3	98	3919.3	71.5
4	137629	2931	3	99	3913.8	71.3
...
67	93519	1807	1	98	2448.6	30.5
68	55935	1807	3	98	2451.6	30.6
69	73080	1806	2	98	2448.8	30.5
70	195573	1805	3	94	2445.0	30.4
71	31552	1800	4	99	2445.0	30.4

72 rows × 6 columns



- total_score의 스케일링 하여 상대 평가
- 30점을 기준으로 빈도수가 많이 줄어들
- 단골 고객 선정 기준을 **total_score점수 30점 이상**으로 설정
- 최종적으로 총 72명의 단골고객 선정 완료!



Pandas' Conclusion

- 데이터 시각화 분석 결과 요약
- 전략 수립 및 실행
- FAQ



데이터 시각화 분석 결과 요약

1. 가장 많이 팔린 품목

- **바나나** 단품과 바나나 한묶음이 가장 많이 팔린 제품으로 확인됨.
흥미로운 점은 상위 10개 품목 중 대부분 **Organic(유기농)** 제품임.

2. 마트 인기 품목 **구매왕** 찾기

- 가장 많은 바나나를 구매한 고객과, 가장 많이 재구매한 고객을 찾음.
- 제품 수량 정보가 없어서 바나나 구매 횟수와 재구매 횟수가 상당히 유사하여 두 지표의 비교는 무의미함.





데이터 시각화 분석 결과 요약

3. 유기농 제품을 파헤쳐보자!

- 유기농제품이 고객에게 끼치는 영향이 매우 크다는 것을 발견. 판매 상품 중 유기농 제품은 **10%의 비중**을 차지함.
- 유기농 제품은 전체 판매량 중 **30% 이상**을 차지함.
- 가장 많이 판매된 유기농 제품을 자주 구매한 고객 파악

4. 사고 또 사고! 재구매왕

- 재구매 비율이 높은 상품 중 유의미한 판매 비율을 가진 제품 파악.
- 재주문 비율이 높으면서 판매량이 동시에 높은 제품은 **유제품**으로써, 고객의 니즈가 크게 작용하는 제품군임.





데이터 시각화 분석 결과 요약

5. 마트의 **Best & Worst** 코너

- 상위 판매 품목은 모두 **식품** 카테고리.
- 하위 판매 품목은 주류 및 **비식품** 카테고리가 대다수.
- 가장 판매량이 낮은 카테고리에서도 유기농 제품이 끼치는 영향을 확인하여, **유기농 키워드의 영향력** 확인
- 판매하는 품목의 종류는 매우 다양하지만 고객들이 가장 선호하는 제품군은 **농작물을 비롯한 식품류**임.





데이터 시각화 분석 결과 요약

6. 마트에 가장 자주 방문하는 단골을 찾아라!

- 재방문 주기 평균이 짧으면서 재방문 횟수가 높은 고객을 찾아 단골 고객으로 선정.
- 구매 물품 횟수도 고려하여 **종합 점수**를 생성 후 표준화하여 단골 고객 선정
- 방문 횟수를 토대로 단골고객선정을 하려 하였으나 데이터에서 제공하는 부분 중, 포괄 처리되어있는 부분이 있음을 발견 해서 새로운 기준을 한번 더 세움.
- 재방문 데이터에서 30일 이상 넘어가는 경우는 모두 30일로 포괄처리됨
- 재방문 데이터에서 재방문 횟수가 99회가 넘어가는 경우는 모두 99회로 포괄처리됨





데이터 시각화 분석 결과 요약

7. 판매 키워드 분석!

- 총 3천 3백만 여개의 판매 정보 중 제품명을 단어 단위로 분리하여 고객이 구매한 키워드를 워드 클라우드를 이용하여 시각화
- **Organic** 이 가장 많이 사용 됨
- Banana, Apple, Avocado, coconut, Almond, Cheese, Yogurt, Milk, Butter, Water, Chicken 등의 **제품 키워드**
- Large, Free, White, Whole, Natural, Original, Sparkling , & 등의 **형용사 키워드**





Pandas' Conclusion

- 데이터 시각화 분석 결과 요약
- 전략 수립 및 실행
- FAQ



전략 수립 및 실행

1. 마트 인기 품목 **구매왕** 찾기

- **소셜 미디어 챌린지**: 구매왕을 찾는 챌린지를 소셜 미디어에서 주최하여 고객들에게 인기 제품을 소셜 미디어에서 홍보 유도. 가장 많이 구매한 고객을 선정하여 선정된 고객에게 특별한 인센티브나 할인 혜택을 제공하여 다른 고객들도 동참하도록 유도

2. **유기농 제품** 을 파헤쳐보자!

- **유기농 제품 패키지**: 유기농 제품을 한데 모은 특별한 유기농 패키지를 만들어서 구매할 때 추가 혜택을 제공함. 이를 통해 고객은 다양한 유기농 제품을 경험하고, 마트에서 유기농 제품을 주목하게 할 수 있음.
- **유기농 농장 투어**: 마트와 협력하는 유기농 농장에서 유기농 제품 생산 과정을 보여주는 투어를 개최함. 이로써 고객들은 유기농 제품의 가치를 느끼고, 환경 친화적인 면을 직접 보고 체험하게 됨.





전략 수립 및 실행

3. 사고 또 사고! 재구매왕

- **구독 서비스:** 재구매율이 높은 상품에 대해 구독 서비스를 도입함. 고객은 자주 사는 상품을 자동으로 주문하고 특별 할인 혜택을 받을 수 있음.
- **커뮤니티 기반 리뷰 및 팁 공유:** 온라인 커뮤니티 활용하여 재구매율이 높은 상품에 대한 고객들은 제품 리뷰 및 사용 팁을 공유하고, 상품에 대한 경험을 나누는 공간을 제공함. 이로써 브랜드 로열티를 높이고, 상품에 대한 고객 만족도를 향상할 수 있음.

4. 마트의 **Best & Worst** 코너

- **시각적 경험 강화:** 사이트 내에서 Best 코너와 Worst 코너를 더욱 시각적으로 강조하고, 해당 상품에 대한 상세 정보와 판촉 이벤트를 시각적으로 제공함.
- **코너 경품 추천:** Best 코너와 Worst 코너에서 구매한 고객들을 대상으로 경품 추천 이벤트를 주최하여 더 다양한 상품을 판매할 수 있도록 유도합니다.





전략 수립 및 실행

5. 마트에 가장 자주 방문하는 **단골**을 찾아라!

- **맞춤형 상품 추천**: 단골 고객의 이전 구매 기록을 기반으로 맞춤형 상품 추천을 제공합니다. 이를 통해 고객은 더욱 편리하게 쇼핑을 할 수 있음.
- **VIP 전용 이벤트 및 선점 판매**: VIP 멤버들을 위한 독점 이벤트를 주최하고, 특정 상품의 선점 판매 기회를 제공하여 특별한 구매 경험을 제공함

6. 판매 **키워드** 분석!

- **키워드 주도 마케팅**: 가장 인기 있는 키워드에 맞춘 제품 라인업을 개발하고, 해당 키워드와 관련된 캠페인을 진행하여 인기 제품과 관련된 트렌드에 민감하게 대응함.
- **소셜 미디어 활용**: 특정 키워드를 활용한 해시태그 및 캠페인을 소셜 미디어에 활발하게 공유하여 고객 참여를 유도하고 제품 홍보를 강화함.





Pandas' Conclusion

- 데이터 시각화 분석 결과 요약
- 전략 수립 및 실행
- FAQ



FAQ

1. 단골 선정기준 **가중치 설정** 이유?

- 재방문 주기(1.5) : 자주 온다 -> 구매를 많이 한다 (가장 중요)
- 상품 구매 횟수(1.3) : 가장 직관적인 선정기준
- 재방문 횟수(1): 단순한 방문 횟수로는 다른 두 가지 기준에 비해 중요도가 떨어짐

2. 왜 **유기농**에 꽂혔나?

- 데이터를 어느 방향으로 분석하든 유기농이라는 키워드는 빠지지 않음
- 심지어 대분류의 alcohol(주류)품목에 마저 유기농 키워드가 들어감





FAQ

3. 해당 데이터를 분석하면서 **이상한 점**은?

- 상품 판매 데이터에서 가장 중요하다고 할 수 있는 **상품 가격, 고객 결제금액, 판매 수량**이 없었음
- 시간이 더 주어졌다면 판매 수량까지는 도출해낼 수 있을 것 같으나 난이도가 상당히 높을 것 같다.

4. **Hexbin Plot**의 사용 이유?

- 데이터의 밀도를 시각적으로 표현할 수 있기 때문
- 산점도는 데이터 포인트가 겹쳐서 보이거나 밀집된 영역의 밀도를 구별하기 어려울 수 있지만 Hexbin Plot은 이러한 문제를 해결하기 위해 사용 됨.

