

Introduction and Objectives

This project aims to leverage data to improve restaurant operations and customer experience.

1 How To Clean Data

To clean data in Python, we can use libraries like Pandas to handle tasks such as removing duplicates, filling or dropping missing values, and correcting inconsistent formatting.

2 Understanding Customer Preferences

Analyze customer feedback, orders, and spending patterns to understand what customers like and dislike.

3 Optimizing Operations

Identify inefficiencies in staffing, inventory management, and kitchen processes to improve service and reduce costs.



Exploratory Data Analysis

Initial analysis helps to understand the data structure and identify potential patterns and relationships.

Descriptive Statistics

Calculate key metrics like average order value, customer churn rate, and inventory turnover.

Data Visualization

Create charts and graphs to visualize trends, relationships, and outliers in the data.

Data Cleaning and Transformation

Address missing values, inconsistent data entries, and other data quality issues.

How To clean Data

- Fill Missing Values
- Remove Rows or Columns With Missing Values
- Naming Conventions
- Remove Duplicates
- Numeric Outliers
- Remove Spaces In Character



NOW Let's see cleaning stage in code

First we Need to read Data

Project.ipynb

File Edit View Insert Runtime Tools Help Last edited on 24 September

Comment

Share

+ Code + Text

Connect Gemini

[]

```
import numpy as np
import pandas as pd
import seaborn as sb
import seaborn as sns

import matplotlib.pyplot as plt
```

#data = pd.read_csv('data_better.csv')

data = pd.read_csv('/content/data_better.csv')

data.head(100)

	order_id	date	item_name	item_type	item_price	quantity	transaction_amount	transaction_type	received_by	time_of_sale
0	1	07-03-2022	Aalopuri	Fastfood	20	13	260	NaN	Mr.	Night
1	2	8/23/2022	Vadapav	Fastfood	20	15	300	Cash	Mr.	Afternoon
2	3	11/20/2022	Vadapav	Fastfood	20	1	20	Cash	Mr.	Afternoon
3	4	02-03-2023	Sugarcane juice	Beverages	25	6	150	Online	Mr.	Night
4	5	10-02-2022	Sugarcane juice	Beverages	25	8	200	Online	Mr.	Evening
...
95	96	8/22/2022	Sandwich	Fastfood	60	15	900	NaN	Mr.	Morning
96	97	3/29/2023	Cold coffee	Beverages	40	13	520	Online	Mr.	Evening
97	98	10/04/2022	Vadapav	Fastfood	20	1	20	Online	Mr.	Morning

First step
identify
datatype of
columns

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1000 entries, 0 to 999  
Data columns (total 10 columns):  
#   Column          Non-Null Count  Dtype    
---  ---            -  
0   order_id        1000 non-null   int64    
1   date            1000 non-null   object    
2   item_name       1000 non-null   object    
3   item_type       1000 non-null   object    
4   item_price      1000 non-null   int64     
5   quantity        1000 non-null   int64     
6   transaction_amount 1000 non-null   int64     
7   transaction_type  893 non-null    object    
8   received_by     1000 non-null   object    
9   time_of_sale    1000 non-null   object    
dtypes: int64(4), object(6)  
memory usage: 78.2+ KB
```

Second step
Check
columns
name

▶ data.columns

➔ Index(['order_id', 'date', 'item_name', 'item_type', 'item_price', 'quantity',
'transaction_amount', 'transaction_type', 'received_by',
'time_of_sale'],
dtype='object')

Change “/”
by “-” to sort
date in
correct way

```
[ ] data['date'] = data['date'].str.replace('/', '-')  
  
data['date'].head()
```

	date
0	07-03-2022
1	8-23-2022
2	11-20-2022
3	02-03-2023
4	10-02-2022

dtype: object

Check Nulls in Columns

data.isnull().sum()

	0
order_id	0
date	0
item_name	0
item_type	0
item_price	0
quantity	0
transaction_amount	0
transaction_type	107
received_by	0
time_of_sale	0

dtype: int64

Fill Missing Value



```
[ ] # Assuming 'data' is your DataFrame
    # Calculate the most frequent value in the 'transaction_type' column
    most_frequent_value = data['transaction_type'].value_counts().idxmax()

    # Fill NaN values with this most frequent value
    data['transaction_type'].fillna(most_frequent_value, inplace=True)

    # Check the result
    print(data['transaction_type'].value_counts())
```

```
transaction_type
Cash      583
Online    417
Name: count, dtype: int64
```

NOW we Need To
Calculate new column
To Calculate Total cost

New Column

```
[ ] # Calculate total cost
data['total_cost'] = data['item_price'] * data['quantity']
```

data.head()

	date	item_name	item_type	item_price	quantity	transaction_type	received_by	time_of_sale	total_cost
0	07-03-2022	Aalopuri	Fastfood	20	13	NaN	Mr.	Night	260
1	8-23-2022	Vadapav	Fastfood	20	15	Cash	Mr.	Afternoon	300
2	11-20-2022	Vadapav	Fastfood	20	1	Cash	Mr.	Afternoon	20
3	02-03-2023	Sugarcane juice	Beverages	25	6	Online	Mr.	Night	150
4	10-02-2022	Sugarcane juice	Beverages	25	8	Online	Mr.	Evening	200

Check value count

Code



```
[ ] data['item_name'].value_counts()
```



count	
item_name	
Cold coffee	161
Sugarcane juice	153
Panipuri	150
Frankie	139
Aalopuri	134
Vadapav	134
Sandwich	129

dtype: int64

```
[ ] data['item_type'].value_counts()
```



count	
item_type	
Fastfood	686
Beverages	314

dtype: int64



```
[ ] data['transaction_type'].value_counts()
```



count	
transaction_type	
Cash	476
Online	417

dtype: int64



```
[ ] data['item_price'].value_counts()
```




count	
item_price	
20	418
40	161
25	153
50	139
60	129

dtype: int64

Code


```
[ ] data['time_of_sale'].value_counts()
```



count	
time_of_sale	
Night	205
Afternoon	205
Evening	201
Midnight	199
Morning	190

dtype: int64


```
[ ] data['received_by'].value_counts()
```



count	
received_by	
Mr.	512
Mrs.	488

dtype: int64

```
[ ] data['quantity'].value_counts()
```



count	
quantity	
13	86
10	81
7	76
15	74
14	73
1	71
2	70
6	69
5	61
12	61
3	60
8	57
9	57
4	55
11	49

We Need to understand customer Preference!

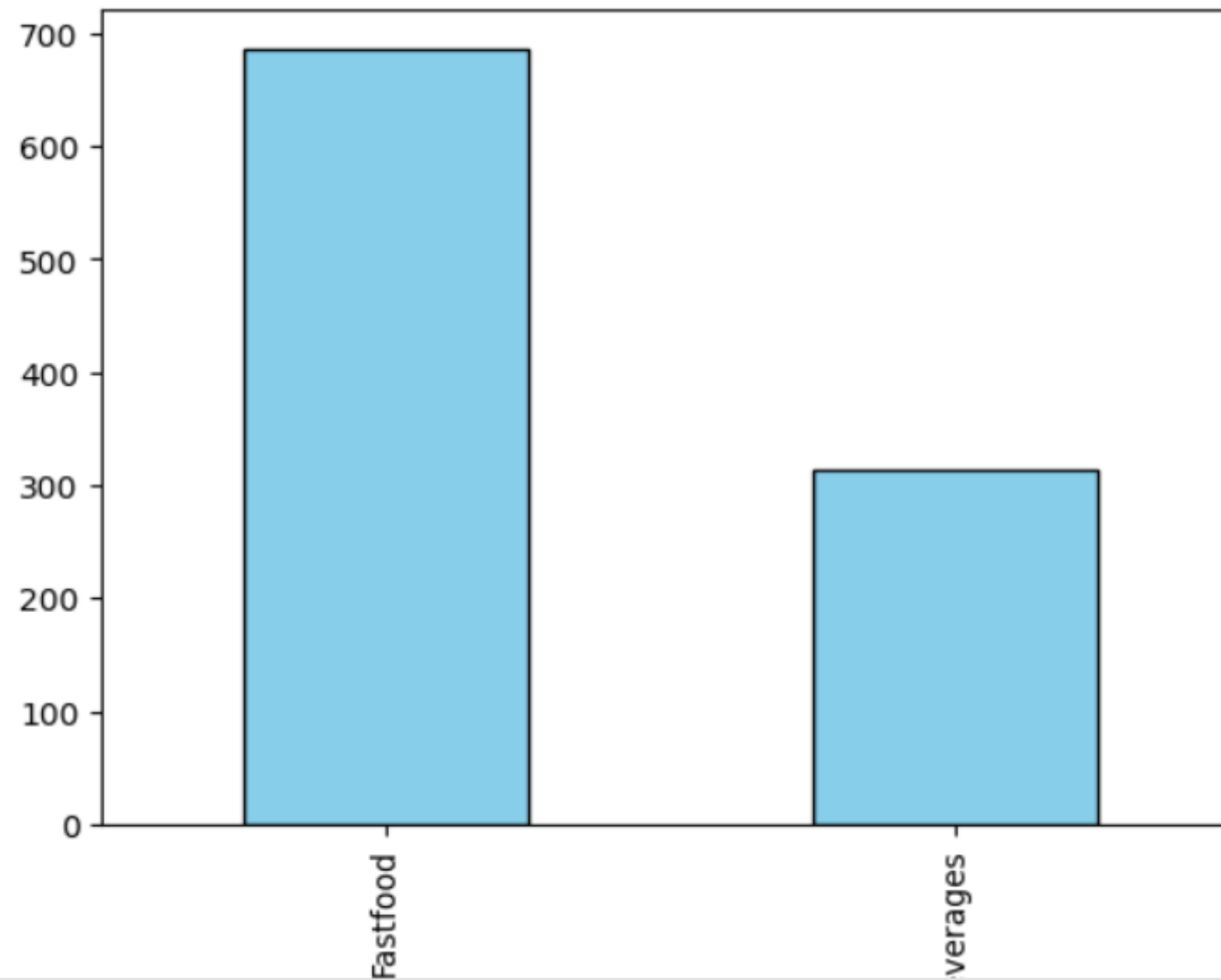


What is the Distribution of Item Types?

```
[ ] itcnt = data['item_type'].value_counts()
```

```
[ ] itcnt.plot(kind='bar', color='skyblue', edgecolor='black')
```

↔ <Axes: xlabel='item_type'>

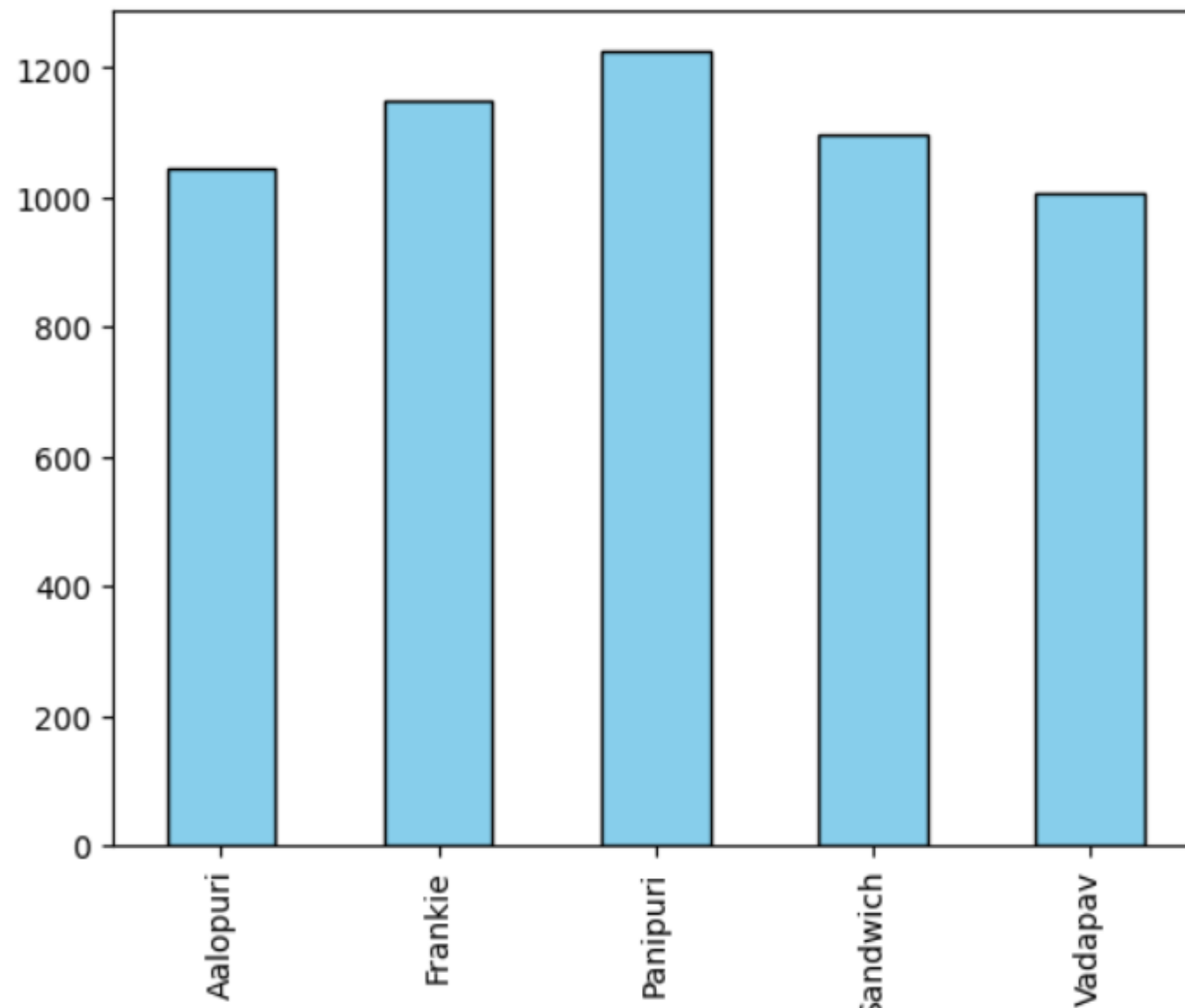


Which Fast Food Items Have the Highest Quantity sold ?

```
[ ] itname_quant = data[data['item_type']=='Fastfood'].groupby('item_name')['quantity'].sum()
```

```
▶ itname_quant.plot(kind='bar', color='skyblue', edgecolor='black') # for fast food only
```

↔ <Axes: xlabel='item_name'>

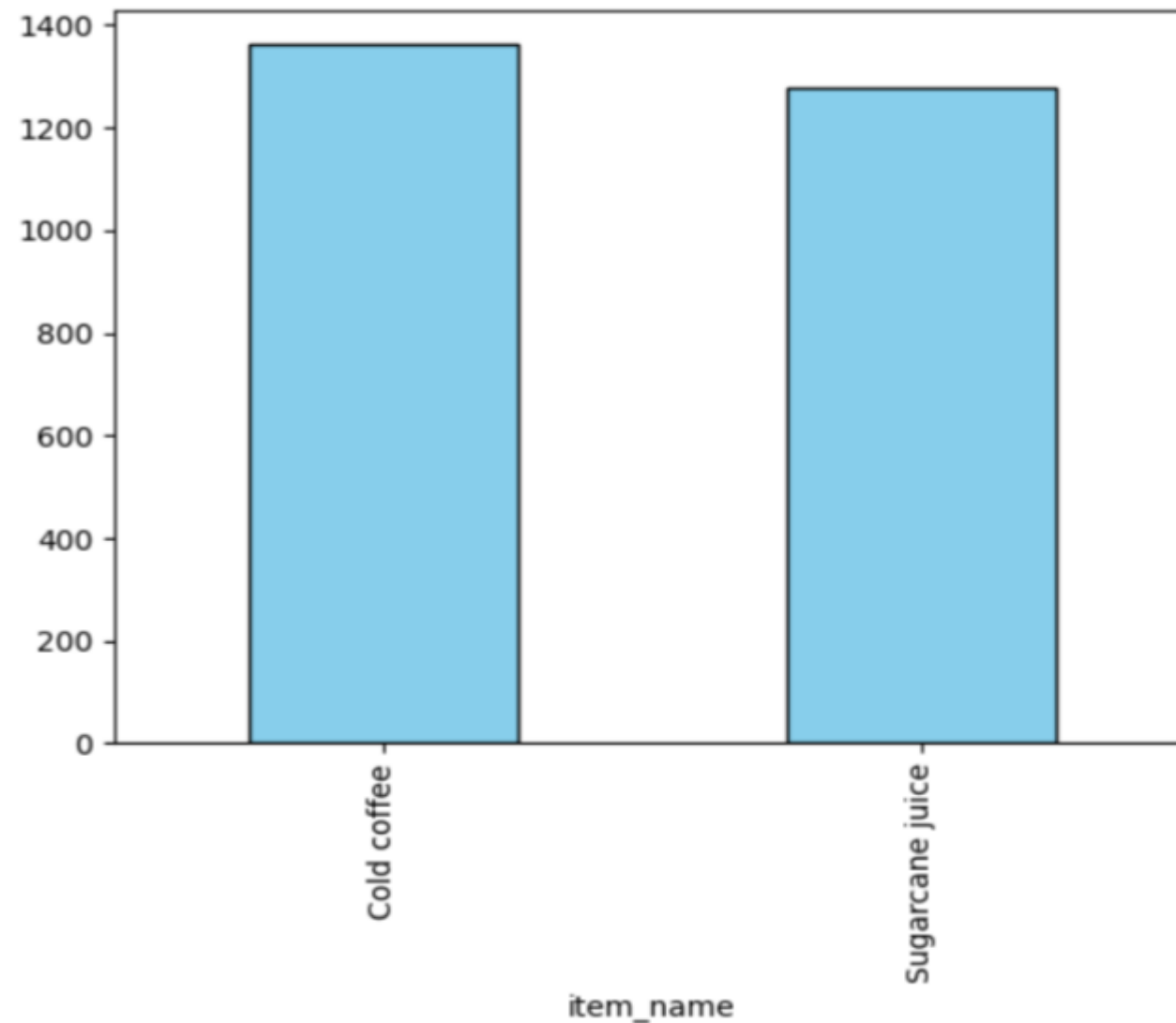


Which Beverage Items Have the Highest Quantity Sold?

```
[ ] itname_quant = data[data['item_type']=='Beverages'].groupby('item_name')['quantity'].sum()
```

```
▶ itname_quant.plot(kind='bar', color='skyblue', edgecolor='black') # for beverages only
```

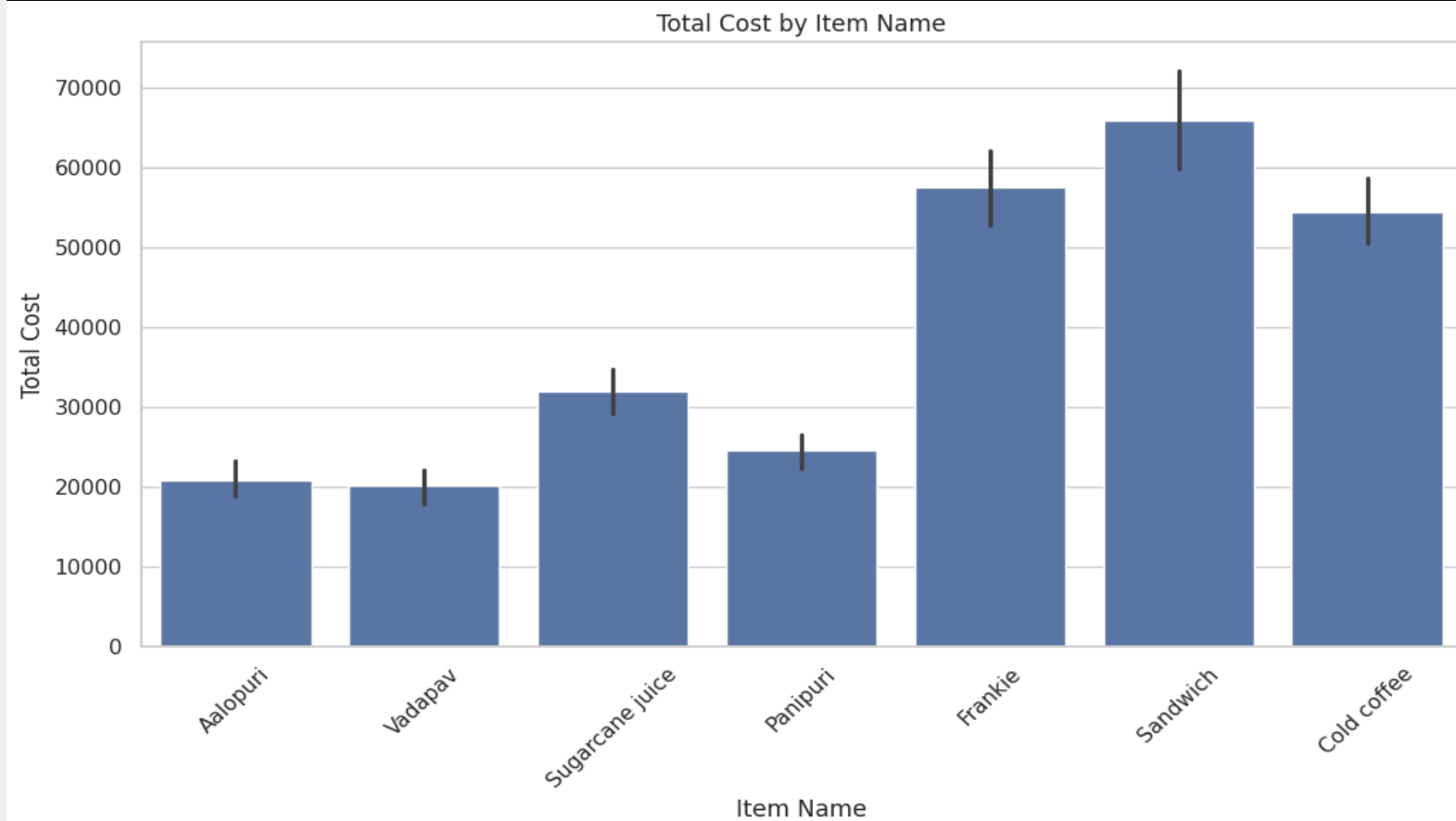
```
↔ <Axes: xlabel='item_name'>
```



What is Most selling item ?

```
# Set the aesthetic style of the plots
sns.set(style="whitegrid")

# Create a bar plot of revenue by item_name
plt.figure(figsize=(12, 6))
sns.barplot(x='item_name', y='total_cost', data=data, estimator=sum)
plt.xticks(rotation=45)
plt.title('Total Cost by Item Name')
plt.xlabel('Item Name')
plt.ylabel('Total Cost')
plt.show()
```

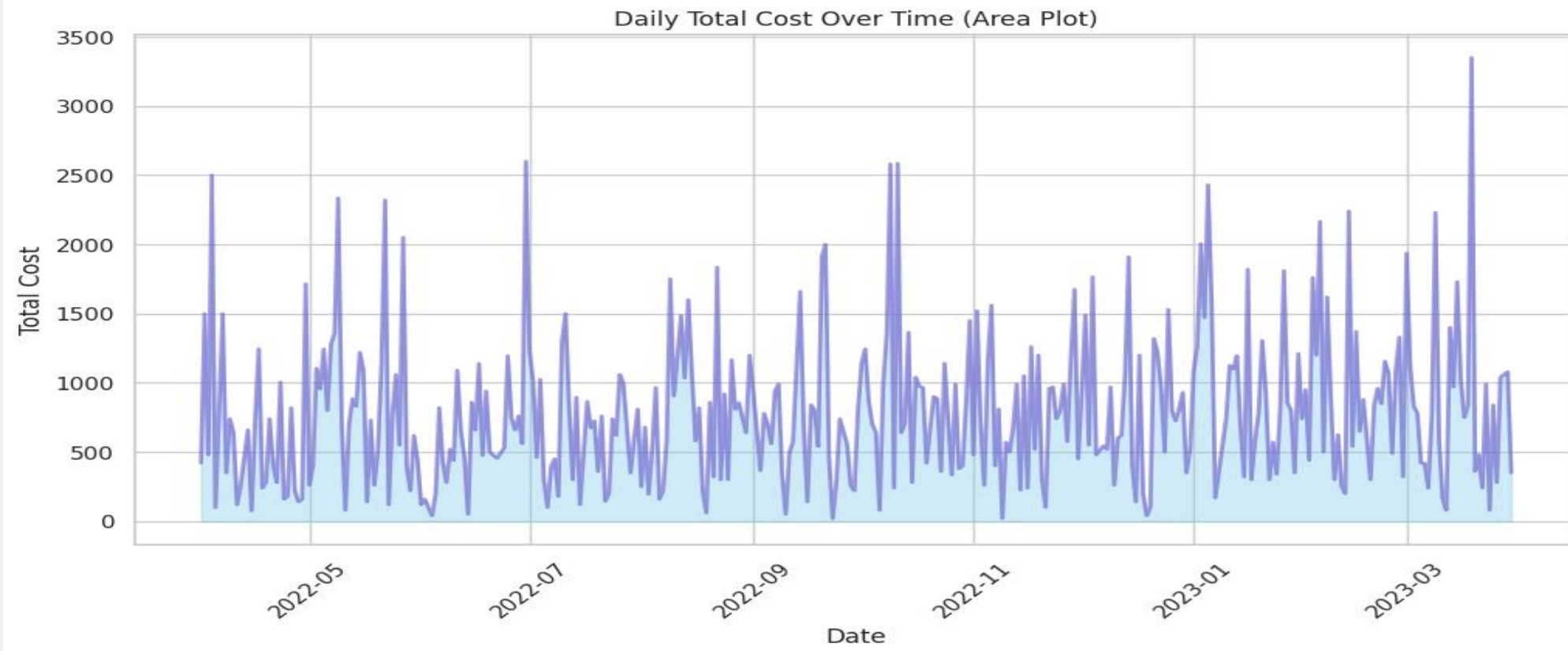


When were the highest total costs recorded?

```
# Create an area plot of total cost over time
# Ensure 'date' is in datetime format
data['date'] = pd.to_datetime(data['date'])

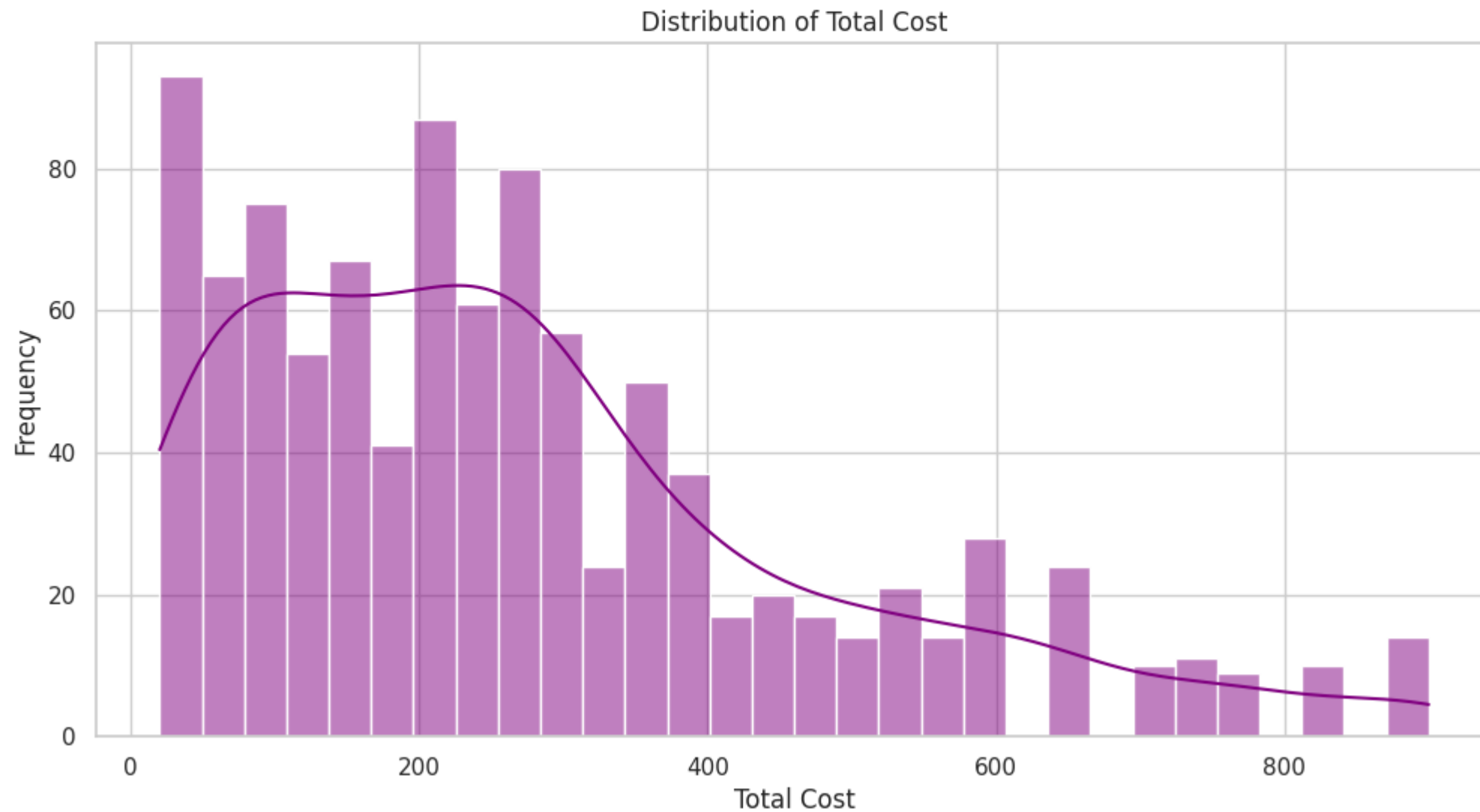
# Aggregate total cost by date
daily_revenue = data.groupby('date')['total_cost'].sum().reset_index()

plt.figure(figsize=(12, 6))
plt.fill_between(daily_revenue['date'], daily_revenue['total_cost'], color='skyblue', alpha=0.4)
plt.plot(daily_revenue['date'], daily_revenue['total_cost'], color='Slateblue', alpha=0.6, linewidth=2)
plt.title('Daily Total Cost Over Time (Area Plot)')
plt.xlabel('Date')
plt.ylabel('Total Cost')
plt.xticks(rotation=45)
plt.show()
```



Distribution of Total Cost?

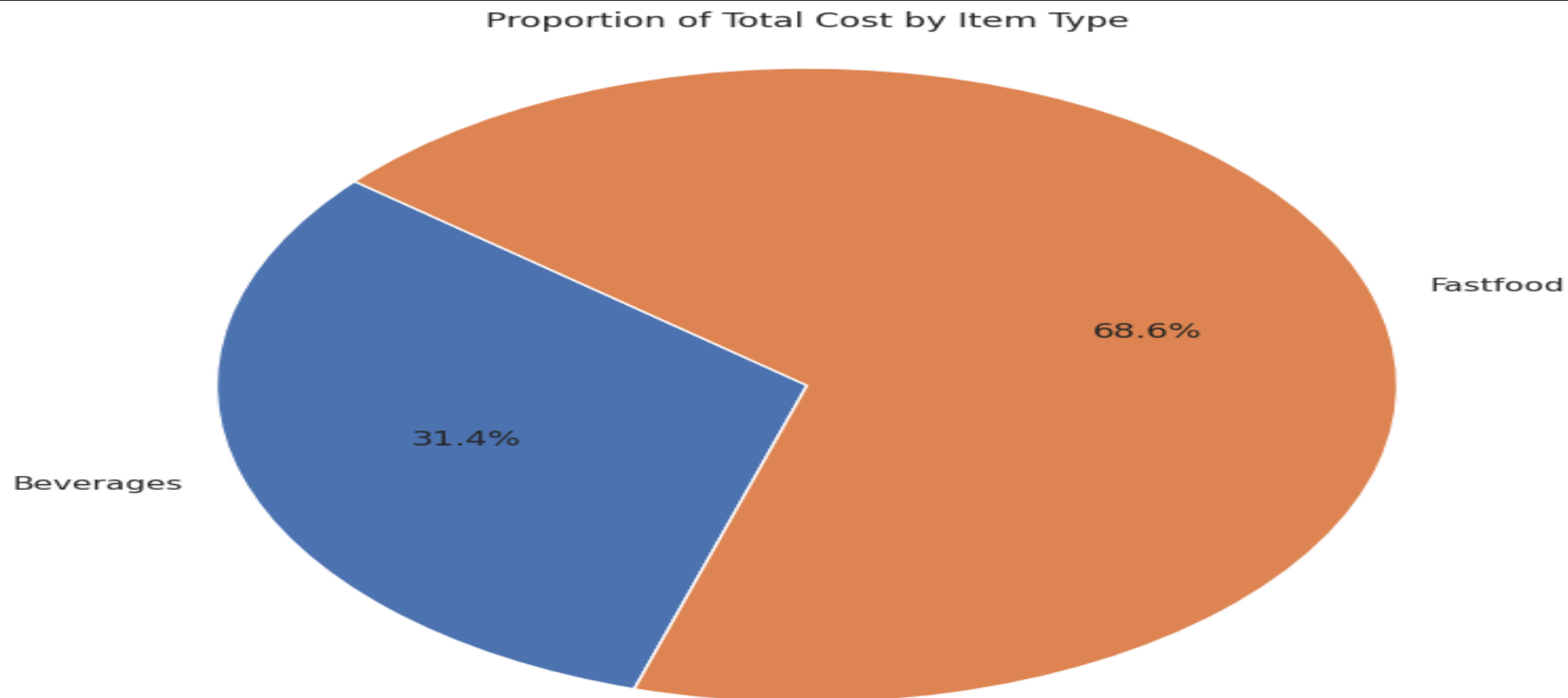
```
# Create a histogram of total cost
plt.figure(figsize=(12, 6))
sns.histplot(data['total_cost'], bins=30, kde=True, color='purple')
plt.title('Distribution of Total Cost')
plt.xlabel('Total Cost')
plt.ylabel('Frequency')
plt.show()
```



What is the Distribution of item type ?

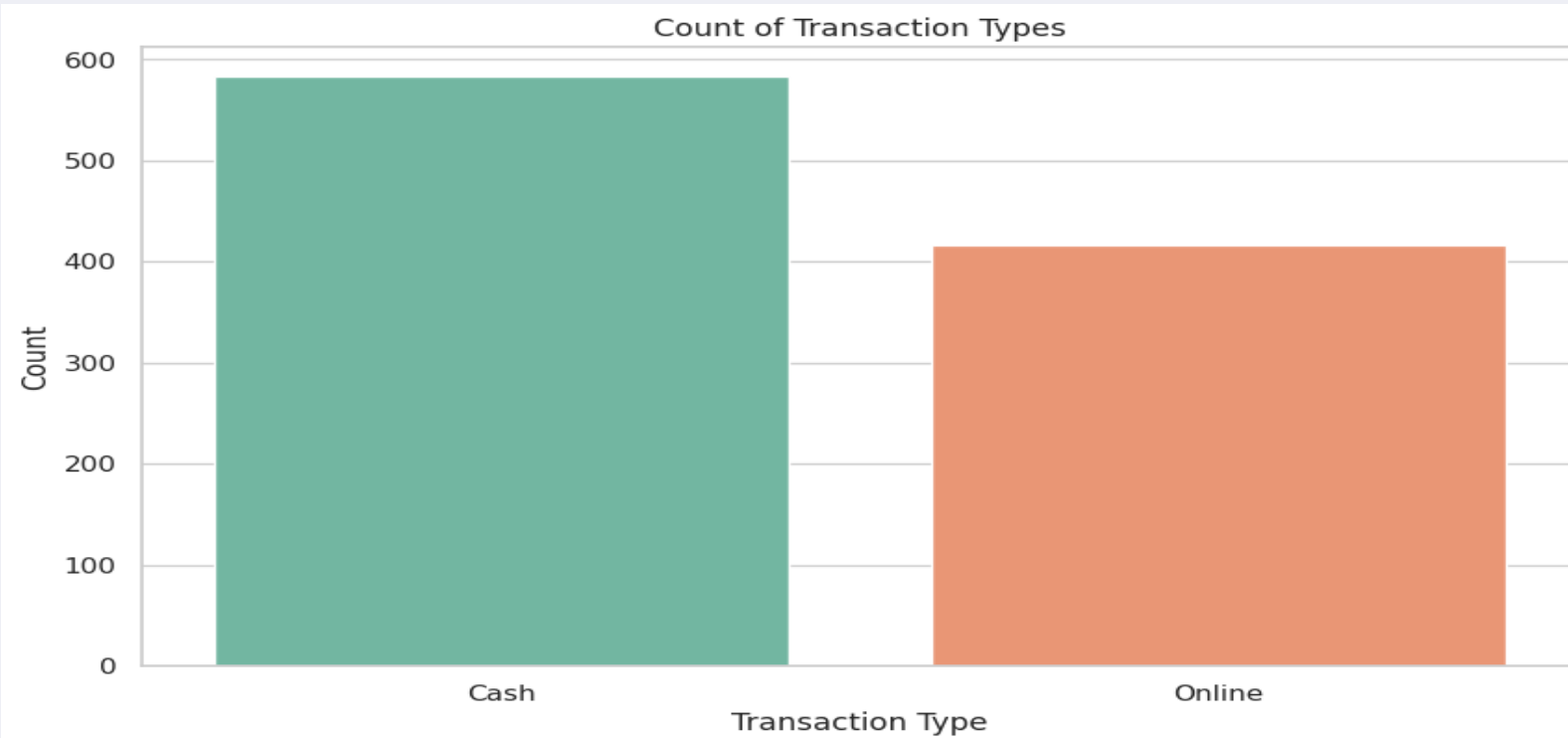
```
# Aggregate total cost by item type
item_costs = data.groupby('item_type')['total_cost'].sum()

# Create a pie chart
plt.figure(figsize=(10, 8))
plt.pie(item_costs, labels=item_costs.index, autopct='%1.1f%%', startangle=140)
plt.title('Proportion of Total Cost by Item Type')
plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```



Which transaction type is used more frequently?

```
#Count Plot of Transaction Types  
plt.figure(figsize=(10, 6))  
sns.countplot(x='transaction_type', data=data, palette='Set2')  
plt.title('Count of Transaction Types')  
plt.xlabel('Transaction Type')  
plt.ylabel('Count')  
plt.show()
```



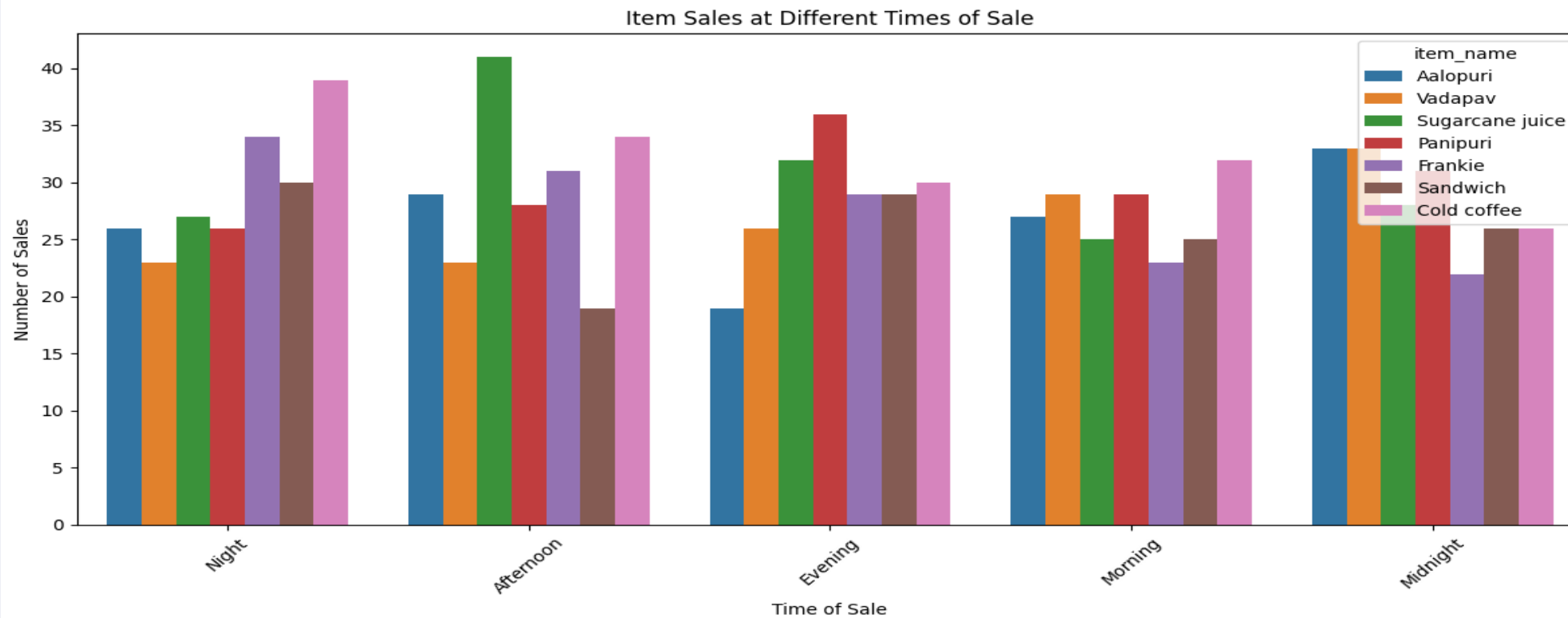
At what time of day are items sold the most frequently?

```
# Create a count plot to visualize the frequency of items sold at different times of sale
plt.figure(figsize=(12, 6))
sns.countplot(data=data, x='time_of_sale', hue='item_name')

# Rotate x-axis labels for better readability if needed
plt.xticks(rotation=45)

# Add title and labels
plt.title('Item Sales at Different Times of Sale')
plt.xlabel('Time of Sale')
plt.ylabel('Number of Sales')

# Show the plot
plt.tight_layout()
plt.show()
```



Conclusion





How to optimize operation

Data analysis can identify operational inefficiencies and suggest improvements.



Wait Times

Analyze wait times to optimize staffing and improve customer service.



Ingredient Usage

Track ingredient usage to reduce waste and optimize inventory management.



Labor Costs

Analyze labor costs to optimize staffing schedules and improve productivity.



Turnover Rate

Track employee turnover rate to identify potential issues and improve employee retention.



Recommendations and Action Plan

Data insights are translated into actionable recommendations to improve restaurant operations and customer experience.

Improve Menu Offerings

Based on customer preferences and sales trends, refine menu offerings and introduce new items.

Increase Quantity of product

Increase most sell product according to analysis

Optimize Marketing Campaigns

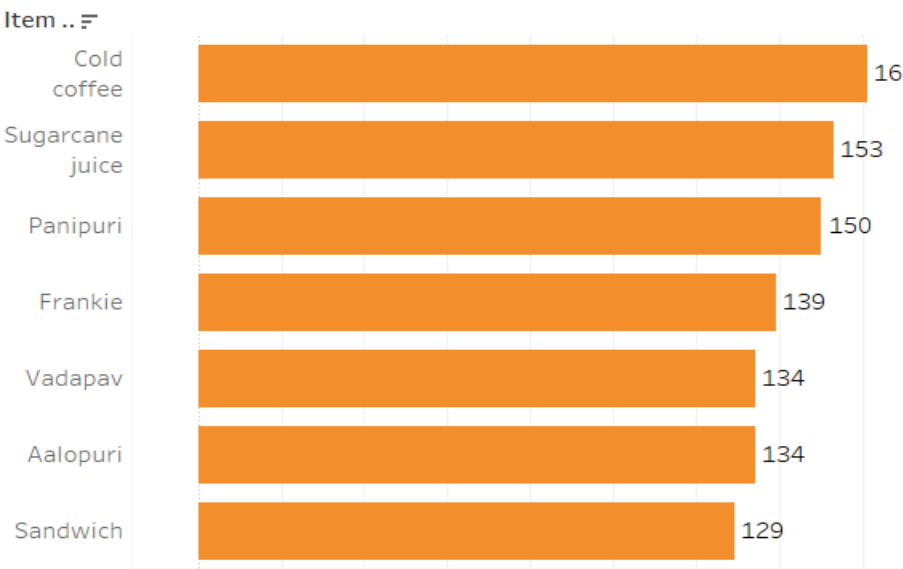
Develop targeted marketing campaigns based on customer segmentation and preferences.

Increase Operational Efficiency

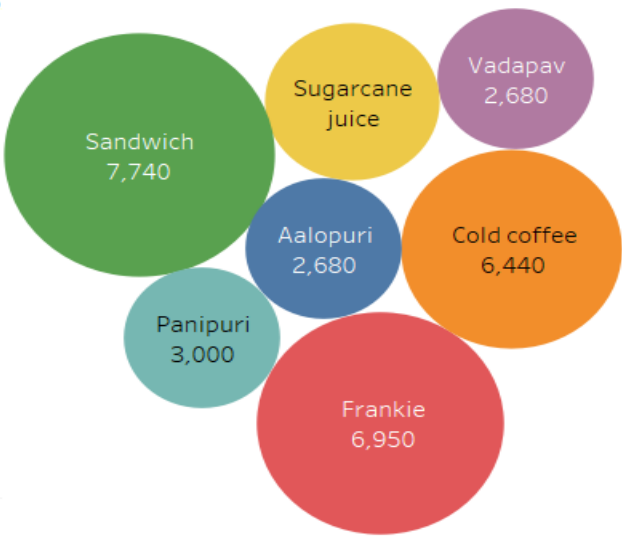
Streamline kitchen processes, optimize inventory management, and implement cost-saving measures.

Restaurant Dashboard

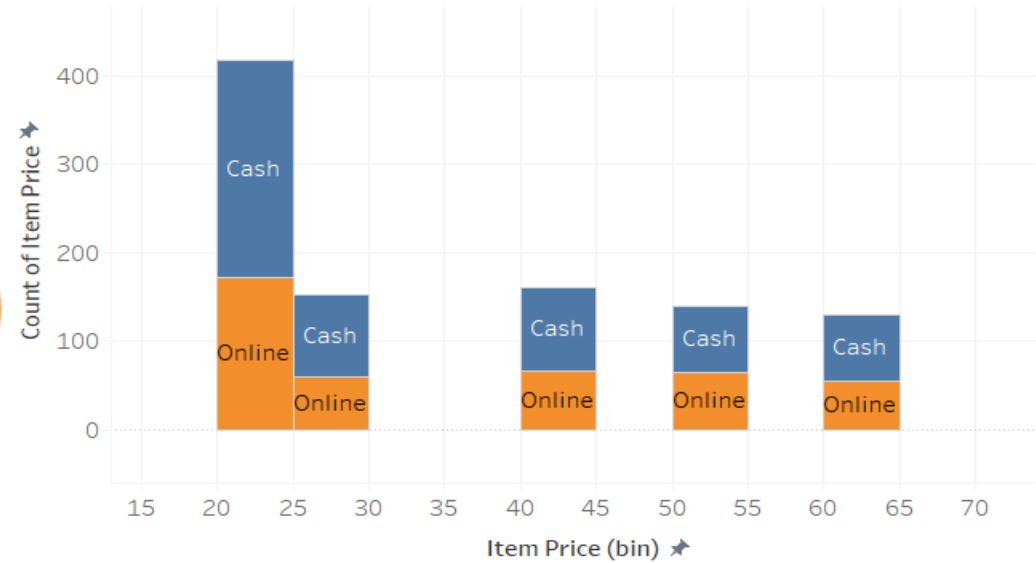
Item Name Sort By value Count



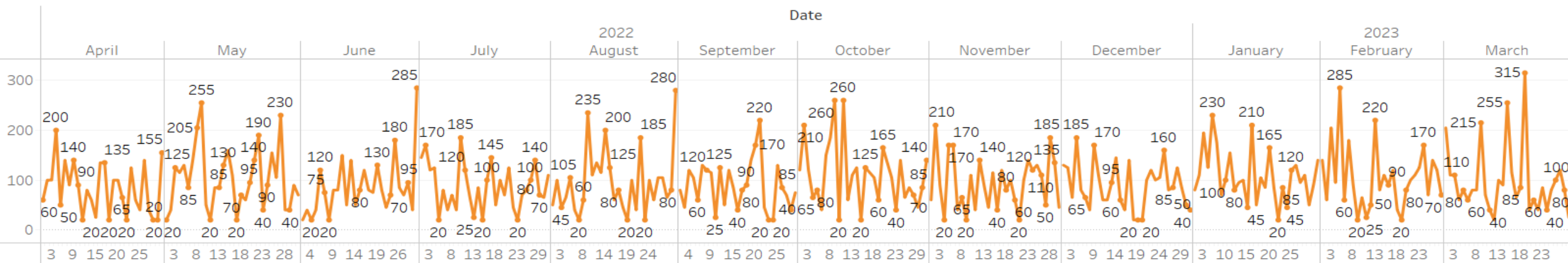
Distribution of Item Name with Item Price



Separation Price Based Transaction Type



Distribution Date Based on Price



Thanks !