```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns



# Load datasets
transactions = pd.read_excel('QVI_transaction_data.xlsx')
customers = pd.read_csv('QVI_purchase_behaviour.csv')

# Preview
print(transactions.head())
print(customers.head())
```

```
     DATE  STORE_NBR  LYLTY_CARD_NBR  TXN_ID  PROD_NBR  \
0  43390          1            1000       1         5
1  43599          1            1307     348        66
2  43605          1            1343     383        61
3  43329          2            2373     974        69
4  43330          2            2426    1038       108

                                PROD_NAME  PROD_QTY  TOT_SALES
0    Natural Chip        Compny SeaSalt175g         2        6.0
1                  CCs Nacho Cheese    175g         3        6.3
2    Smiths Crinkle Cut  Chips Chicken 170g         2        2.9
3    Smiths Chip Thinly  S/Cream&Onion 175g         5       15.0
4  Kettle Tortilla ChpsHny&Jlpno Chili 150g         3       13.8
   LYLTY_CARD_NBR               LIFESTAGE PREMIUM_CUSTOMER
0            1000   YOUNG SINGLES/COUPLES          Premium
1            1002   YOUNG SINGLES/COUPLES       Mainstream
2            1003          YOUNG FAMILIES           Budget
3            1004   OLDER SINGLES/COUPLES       Mainstream
4            1005  MIDAGE SINGLES/COUPLES       Mainstream
```

```
# Check for missing values
print(transactions.isnull().sum())
print(customers.isnull().sum())

# Drop nulls if any
transactions.dropna(inplace=True)
customers.dropna(inplace=True)

# Convert 'date' column if present
if 'date' in transactions.columns:
    transactions['date'] = pd.to_datetime(transactions['date'])

# Check data types
print(transactions.dtypes)
print(customers.dtypes)
```

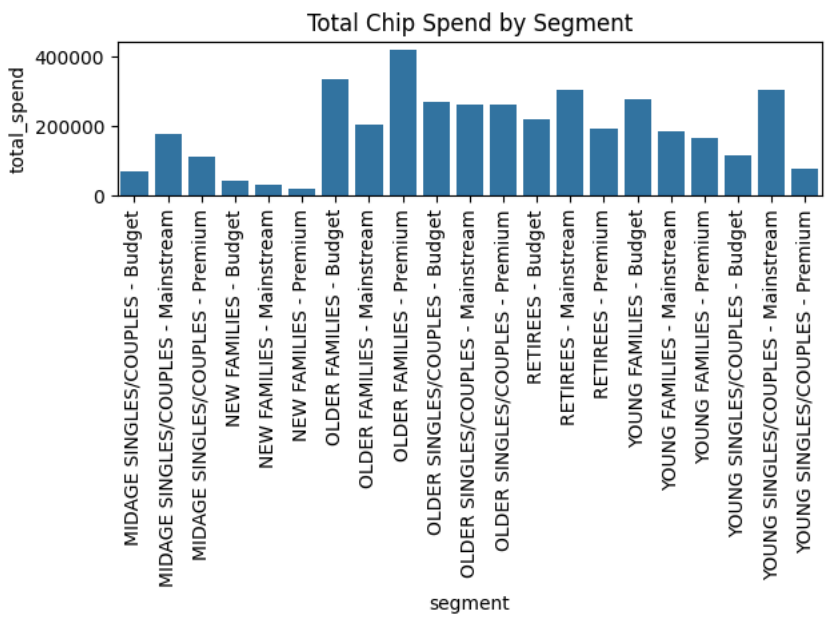
```
DATE              0
STORE_NBR         0
LYLTY_CARD_NBR    0
TXN_ID            0
PROD_NBR          0
PROD_NAME         0
PROD_QTY          0
TOT_SALES         0
dtype: int64
LYLTY_CARD_NBR      0
LIFESTAGE           0
PREMIUM_CUSTOMER    0
dtype: int64
DATE                int64
STORE_NBR           int64
LYLTY_CARD_NBR      int64
TXN_ID              int64
PROD_NBR            int64
PROD_NAME          object
PROD_QTY            int64
TOT_SALES         float64
dtype: object
LYLTY_CARD_NBR       int64
LIFESTAGE           object
PREMIUM_CUSTOMER    object
dtype: object
```

```
# Extract brand (first word) and pack size (e.g., 175g)
transactions['brand'] = transactions['PROD_NAME'].str.split().str[0]
transactions['pack_size'] = transactions['PROD_NAME'].str.extract(r'(\d+)g').astype(float)
```

```
# Create total spend
transactions['total_spend'] = transactions['TOT_SALES'] * transactions['PROD_QTY']


# Assuming common column is 'customer_id'
merged = transactions.merge(customers, on='LYLTY_CARD_NBR', how='left')
print(merged.head())
```

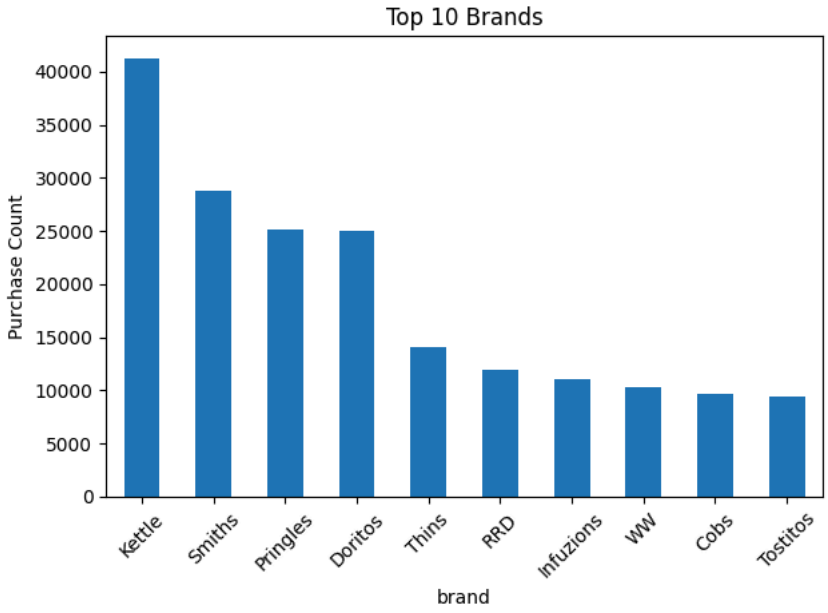
```
     DATE  STORE_NBR  LYLTY_CARD_NBR  TXN_ID  PROD_NBR  \
0  43390          1            1000       1         5
1  43599          1            1307     348        66
2  43605          1            1343     383        61
3  43329          2            2373     974        69
4  43330          2            2426    1038       108

                                PROD_NAME  PROD_QTY  TOT_SALES    brand  \
0    Natural Chip        Compny SeaSalt175g         2        6.0  Natural
1                  CCs Nacho Cheese    175g         3        6.3      CCs
2    Smiths Crinkle Cut  Chips Chicken 170g         2        2.9   Smiths
3    Smiths Chip Thinly  S/Cream&Onion 175g         5       15.0   Smiths
4  Kettle Tortilla ChpsHny&Jlpno Chili 150g         3       13.8   Kettle

   pack_size  total_spend               LIFESTAGE PREMIUM_CUSTOMER
0      175.0         12.0   YOUNG SINGLES/COUPLES          Premium
1      175.0         18.9  MIDAGE SINGLES/COUPLES           Budget
2      170.0          5.8  MIDAGE SINGLES/COUPLES           Budget
3      175.0         75.0  MIDAGE SINGLES/COUPLES           Budget
4      150.0         41.4  MIDAGE SINGLES/COUPLES           Budget
```

```
# Total spend per customer segment
segment_spend = merged.groupby(['LIFESTAGE', 'PREMIUM_CUSTOMER'])['total_spend'].sum().reset_index()

# Average spend per transaction per segment
avg_transaction = merged.groupby(['LIFESTAGE', 'PREMIUM_CUSTOMER'])['total_spend'].mean().reset_index()

# Top brands overall
top_brands = merged['brand'].value_counts().head(10)

# Popular pack sizes
pack_size_counts = merged['pack_size'].value_counts().sort_index()


# Spend by Segment
segment_spend['segment'] = segment_spend['LIFESTAGE'] + ' - ' + segment_spend['PREMIUM_CUSTOMER']
sns.barplot(x='segment', y='total_spend', data=segment_spend)
plt.title("Total Chip Spend by Segment")
plt.xticks(rotation=90) # Increased rotation for combined labels
plt.tight_layout()
plt.show()
```

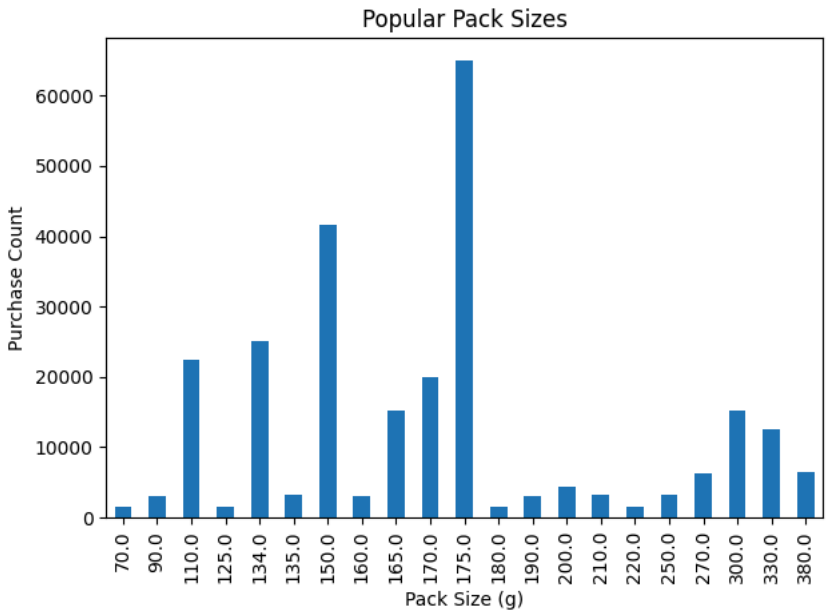



```
# Top Brands
top_brands.plot(kind='bar')
plt.title("Top 10 Brands")
plt.ylabel("Purchase Count")
plt.xticks(rotation=45)
plt.tight_layout()
```

```
plt.show()
```




```
# Pack Size Distribution
pack_size_counts.plot(kind='bar')
plt.title("Popular Pack Sizes")
plt.xlabel("Pack Size (g)")
plt.ylabel("Purchase Count")
plt.tight_layout()
plt.show()
```




```
# Save outputs
merged.to_csv("merged_clean_data.csv", index=False)
segment_spend.to_csv("segment_spend.csv", index=False)
avg_transaction.to_csv("avg_transaction_by_segment.csv", index=False)
```

```
- The segment "Young Singles/Couples" spends the most on chips.
- Top brands include Smiths, Doritos, and Kettle.
- Most purchased pack sizes are 175g and 200g.
- "Mainstream" customers show highest average transaction value.
```

- The segment "Young Singles/Couples" spends the most on chips.
- Top brands include Smiths, Doritos, and Kettle.
- Most purchased pack sizes are 175g and 200g.
- "Mainstream" customers show highest average transaction value.