

W załączeniu otrzymujesz plik dane.csv zawierający dane sprzedażowe pewnego sklepu. Postępując się danymi zawartymi w załączonym pliku dokonaj operacji określonych w poniższych zadaniach.

Zadanie 2.1.

Dodaj kolumnę Margin i oblicz w niej marżę przypadającą na dane zamówienie (przyjmijmy, że marża to różnica pomiędzy wartością zamówienia, a COGS)

Zadanie 2.2.

Dodaj kolumnę CumMargin i oblicz w niej skumulowaną marżę od początku danego roku. Dla lat 2021 i 2022 określ czy (a jeśli tak to w jakich dniach) sklep osiągnął postawione KPI w wysokości 40 000 000.

Zadanie 2.3.

Z posiadanych danych wyodrębnij pliki .csv zawierające dane sprzedażowe z roku 2022, oddzielnie dla każdego kanału sprzedaży, uszeregowane według regionu, daty i godziny.

Zadanie 2.4.

Odkrywasz, że w wyniku błędu dane z niektórych dni załadowały się z błędami:

- kilkakrotnie – w tym przypadku usuń zduplikowane rekordy,
- bez danych o wartości sprzedaży i COGS – w tym przypadku usuń zduplikowane rekordy
- bez danych o COGS – w tym przypadku uzupełnij COGS przyjmując, że COGS dla tych rekordów wynosi 85% wartości sprzedaży

Zadanie 2.5

Oblicz sumę wartości sprzedaży oraz liczbę transakcji po: poszczególnych miesiącach roku 2022, kanale sprzedaży, regionie – oddzielnie dla zamówień złożonych w porze nocnej (przyjmijmy, że pora nocna trwa od godz. 23:00 do godz. 7:00 rano) oraz za dnia.

Zadanie 2.6.

Przedstaw na wizualizacji sumę wartości sprzedaży w poszczególnych miesiącach 2022 roku.

```
import pandas as pd

from tabulate import tabulate

df = pd.read_csv('Dane.csv', sep = ';')
print(tabulate(df.head(), headers='keys', tablefmt='pretty'))

df['COGS'] = df['COGS'].str.replace(",", ".")
df['COGS'] = df['COGS'].str.replace(" ", "")
```

```
df.info()
```

```
df['COGS'] = df['COGS'].astype(float)
df['DateOrder'] = pd.to_datetime(df['DateOrder'])

df['TimeOrder'] = pd.to_datetime(df['TimeOrder']).dt.time
```

```
2.1 df['Margin'] = df['OrderValue'] - df['COGS']
```

```
df['Margin'] = df['Margin'].round(2)
```

```
2.2 df['CumMargin'] = df.groupby(df['DateOrder'].dt.year)['Margin'].cumsum()
```

```
df_2022_achieved_kpi = df.query('CumMargin >= 40000000 and DateOrder.dt.year == 2021')
```

```
first_row_2021_achieved_kpi = df_2021_achieved_kpi.head(1)
```

```
print(first_row_2021_achieved_kpi)
```

```
df_2022_achieved_kpi = df.query('CumMargin >= 40000000 and DateOrder.dt.year == 2022')
```

```
first_row_2022_achieved_kpi = df_2021_achieved_kpi.head(1)
```

```
print(first_row_2022_achieved_kpi)
```

```
2.3 sales_2022 = df.query('DateOrder.dt.year == 2022')
```

```

for channel in sales_2022['OrderChannel'].unique():
    channel_data = sales_2022[sales_2022['OrderChannel'] == channel]

    filename = f"sales_{channel}_2022.csv"

    channel_data.sort_values(by=['OrderRegion', 'DateOrder', 'TimeOrder']).to_csv(filename, index=False)

```

2.4

```

a) df = df.drop_duplicates()
print(df)

b)

df = df.dropna(subset=['OrderValue', 'COGS'])
print(df)

c) df['COGS'].fillna(0.85 * df['OrderValue'], inplace=True)

```

2.5

```

for i,row in df.iterrows():
    if row['TimeOrder'].hour >= 23 or row['TimeOrder'].hour < 7:
        df.loc[i,'DayTime'] = 'Night'
    else:
        df.loc[i,'DayTime'] = 'Day'

```

```
df_2022 = df.query('DateOrder.dt.year == 2022')
```

```
result = df_2022.groupby([df_2022['DateOrder'].dt.month, 'OrderChannel', 'OrderRegion',  
                        'Daytime']).agg(  
    SalesSum=('OrderValue', 'sum'),  
    TransactionCount=('OrderID', 'count')  
).reset_index()  
print(result)
```

2.6

```
import calendar
```

```
import matplotlib.pyplot as plt
```

```
df_2022 = df[df['DateOrder'].dt.year == 2022]
```

```
monthly_sales = df_2022.groupby(df_2022['DateOrder'].dt.strftime('%B'))['OrderValue'].sum()
```

```
months_order = list(calendar.month_name)[1:]
```

```
monthly_sales = monthly_sales.reindex(months_order)
```

```
plt.figure(figsize=(14, 6))
```

```
plt.bar(monthly_sales.index, monthly_sales.values, color='skyblue')
```

```
plt.title('Suma wartości sprzedaży w poszczególnych miesiącach 2022 roku')
```

```
plt.xlabel('Miesiąc')
```

```
plt.ylabel('Suma wartości sprzedaży')
```

```
plt.grid(axis='y')
```

```
plt.show()
```