PIYUSH KUMAR MISHRA

230957212

ROLL NO 70

WEEK 9

In [2]:
```python
import pandas as pd
import numpy as np

# Create a sample dataset
data = {
    'Date': pd.date_range(start='2024-01-01', periods=20, freq='W'),
    'Product Name': np.random.choice(['Widget A', 'Widget B', 'Widget C'], 20),
    'Units Sold': np.random.randint(1, 20, size=20),
    'Revenue': np.random.uniform(100, 1000, size=20).round(2),
    'Region': np.random.choice(['North', 'South', 'East', 'West'], 20),
    'Discount Offered (%)': np.random.uniform(0, 30, size=20).round(2),
    'Salesperson': np.random.choice(['Alice', 'Bob', 'Charlie', 'David'], 20)
}

df = pd.DataFrame(data)

# Calculate Revenue After Discounts
df['Revenue After Discount'] = df['Revenue'] * (1 - df['Discount Offered (%)'] / 100)
print(df)
```

```
         Date Product Name  Units Sold  Revenue Region  Discount Offered (%)  \
0  2024-01-07     Widget B          14   809.91  North                  7.00
1  2024-01-14     Widget A          19   200.50   West                 22.22
2  2024-01-21     Widget B          17   684.44  South                 21.18
3  2024-01-28     Widget A           3   325.69  South                 26.80
4  2024-02-04     Widget B           4   729.73  South                 28.41
5  2024-02-11     Widget B          14   603.22   West                  6.90
6  2024-02-18     Widget B           6   742.65  South                 11.88
7  2024-02-25     Widget B          13   610.45   West                  8.87
8  2024-03-03     Widget C           8   346.44  North                  8.32
9  2024-03-10     Widget C           6   974.03  South                 25.85
10 2024-03-17     Widget C           2   925.73   East                  4.99
11 2024-03-24     Widget B          14   847.99   East                 24.52
12 2024-03-31     Widget B          17   937.64   West                  7.30
13 2024-04-07     Widget B          19   514.51  North                 17.28
14 2024-04-14     Widget C          15   355.25  North                 10.64
15 2024-04-21     Widget B          19   349.53   West                 11.51
16 2024-04-28     Widget B          14   804.63  South                  2.94
17 2024-05-05     Widget B          17   202.10  South                  5.49
18 2024-05-12     Widget B           8   478.21   West                  0.35
19 2024-05-19     Widget C           6   237.60  South                 23.35

   Salesperson  Revenue After Discount
0        David              753.216300
1        Alice              155.948900
2        David              539.475608
3          Bob              238.405080
4          Bob              522.413707
5        Alice              561.597820
6      Charlie              654.423180
7      Charlie              556.303085
8        David              317.616192
9          Bob              722.243245
10         Bob              879.536073
11       Alice              640.062852
12         Bob              869.192280
13     Charlie              425.602672
14     Charlie              317.451400
15       David              309.299097
16     Charlie              780.973878
17       Alice              191.004710
18       Alice              476.536265
19         Bob              182.120400
```

In [3]:
```python
# 1. Top 3 sales transactions with the highest revenue
top_3_revenue = df.nlargest(3, 'Revenue')
print("Top 3 Sales Transactions with Highest Revenue:\n", top_3_revenue)
```

```
Top 3 Sales Transactions with Highest Revenue:
          Date Product Name  Units Sold  Revenue Region  Discount Offered (%)  \
9   2024-03-10     Widget C           6   974.03  South                 25.85
12  2024-03-31     Widget B          17   937.64   West                  7.30
10  2024-03-17     Widget C           2   925.73   East                  4.99

   Salesperson  Revenue After Discount
9          Bob              722.243245
12         Bob              869.192280
10         Bob              879.536073
```

In [4]:
```python
# 2. Units sold for each product
units_sold_per_product = df.groupby('Product Name')['Units Sold'].sum()
print("\nUnits Sold for Each Product:\n", units_sold_per_product)
```

```
Units Sold for Each Product:
 Product Name
Widget A     22
Widget B    176
Widget C     37
Name: Units Sold, dtype: int32
```

In [5]:
```python
# 3. Total revenue after applying discounts
total_revenue_after_discount = df['Revenue After Discount'].sum()
print("\nTotal Revenue After Discounts:\n", total_revenue_after_discount)
```

```
Total Revenue After Discounts:
 10093.422744
```

In [6]:
```python
# 4. Transaction with the highest discount offered
highest_discount = df.loc[df['Discount Offered (%)'].idxmax()]
highest_discount_revenue_after_discount = highest_discount['Revenue After Discount']
print("\nHighest Discount Transaction:\n", highest_discount)
```

```
Highest Discount Transaction:
 Date                      2024-02-04 00:00:00
Product Name                         Widget B
Units Sold                                  4
Revenue                                729.73
Region                                  South
Discount Offered (%)                    28.41
Salesperson                               Bob
Revenue After Discount             522.413707
Name: 4, dtype: object
```

In [7]:
```python
# 5. Salesperson generating the highest total revenue
highest_revenue_salesperson = df.groupby('Salesperson')['Revenue'].sum().idxmax()
print("\nSalesperson with Highest Total Revenue:\n", highest_revenue_salesperson)
```

```
Salesperson with Highest Total Revenue:
 Bob
```

In [8]:
```python
# 6. Average discount offered by each salesperson
avg_discount_per_salesperson = df.groupby('Salesperson')['Discount Offered (%)'].mean()
```

```
print("\nAverage Discount Offered by Each Salesperson:\n", avg_discount_per_salesperson)
```

```
Average Discount Offered by Each Salesperson:
 Salesperson
Alice      11.8960
Bob        19.4500
Charlie    10.3220
David      12.0025
Name: Discount Offered (%), dtype: float64
```

In [9]:
```
# 7. Revenue generated in each region
revenue_per_region = df.groupby('Region')['Revenue'].sum()
print("\nRevenue Generated in Each Region:\n", revenue_per_region)
```

```
Revenue Generated in Each Region:
 Region
East     1773.72
North    2026.11
South    4700.87
West     3179.55
Name: Revenue, dtype: float64
```

In [10]:
```
# 8. Region where Alice generated the highest sales
alice_sales = df[df['Salesperson'] == 'Alice']
alice_highest_sales_region = alice_sales.groupby('Region')['Revenue'].sum().idxmax()
print("\nAlice's Highest Sales Region:\n", alice_highest_sales_region)
```

```
Alice's Highest Sales Region:
 West
```

In [11]:
```
# 9. Product generating the highest revenue per unit sold
df['Revenue Per Unit'] = df['Revenue'] / df['Units Sold']
highest_revenue_per_unit_product = df.loc[df['Revenue Per Unit'].idxmax()]
print("\nHighest Revenue per Unit Sold Product:\n", highest_revenue_per_unit_product)
```

```
Highest Revenue per Unit Sold Product:
 Date                       2024-03-17 00:00:00
Product Name                           Widget C
Units Sold                                    2
Revenue                                  925.73
Region                                     East
Discount Offered (%)                       4.99
Salesperson                                 Bob
Revenue After Discount              879.536073
Revenue Per Unit                       462.865
Name: 10, dtype: object
```

In [13]:
```
# 10. Transactions rated as "High" performance (arbitrarily define as Revenue > 800)
high_performance_transactions = df[df['Revenue'] > 800]
print("\nHigh Performance Transactions Count:\n", len(high_performance_transactions))
```

```
High Performance Transactions Count:
 6
```

In [16]:
```python
# 11. Salesperson sold the most units in North region without offering any discount
north_sales_no_discount = df[(df['Region'] == 'North') & (df['Discount Offered (%)'] == 0)]

if not north_sales_no_discount.empty:
    most_units_north = north_sales_no_discount.groupby('Salesperson')['Units Sold'].sum().idxmax()
    print("\nMost Units Sold in North Region Without Discount:\n", most_units_north)
else:
    print("\nNo sales in North region without discount.")
```

```
No sales in North region without discount.
```

In [17]:
```python
# 12. Average revenue per unit sold in each region for each product
avg_revenue_per_unit_region_product = df.groupby(['Region', 'Product Name']).apply(lambda x: (x['Revenue'] / x['Units Sold']).mean())
print("\nAverage Revenue per Unit Sold in Each Region for Each Product:\n", avg_revenue_per_unit_region_product)
```

```
Average Revenue per Unit Sold in Each Region for Each Product:
 Region  Product Name
East    Widget B         60.570714
        Widget C        462.865000
North   Widget B         42.465094
        Widget C         33.494167
South   Widget A        108.563333
        Widget B         83.166097
        Widget C        100.969167
West    Widget A         10.552632
        Widget B         44.674539
dtype: float64
```

In [18]:
```python
# 13. Salesperson with the highest average revenue after discounts
avg_revenue_after_discount_per_salesperson = df.groupby('Salesperson')['Revenue After Discount'].mean().idxmax()
print("\nSalesperson with Highest Average Revenue After Discounts:\n", avg_revenue_after_discount_per_salesperson)
```

```
Salesperson with Highest Average Revenue After Discounts:
 Bob
```

In [19]:
```python
# 14. Cumulative total revenue over time for each salesperson
cumulative_revenue = df.groupby(['Salesperson', 'Date'])['Revenue'].sum().groupby(level=0).cumsum()
print("\nCumulative Total Revenue Over Time for Each Salesperson:\n", cumulative_revenue)
```

```
Cumulative Total Revenue Over Time for Each Salesperson:
 Salesperson  Date
Alice        2024-01-14     200.50
             2024-02-11     803.72
             2024-03-24    1651.71
             2024-05-05    1853.81
             2024-05-12    2332.02
Bob          2024-01-28     325.69
             2024-02-04    1055.42
             2024-03-10    2029.45
             2024-03-17    2955.18
             2024-03-31    3892.82
             2024-05-19    4130.42
Charlie      2024-02-18     742.65
             2024-02-25    1353.10
             2024-04-07    1867.61
             2024-04-14    2222.86
             2024-04-28    3027.49
David        2024-01-07     809.91
             2024-01-21    1494.35
             2024-03-03    1840.79
             2024-04-21    2190.32
Name: Revenue, dtype: float64
```

In [20]:

```python
# 15. Rank transactions by revenue for each salesperson and find top 2
top_2_transactions_per_salesperson = df.groupby('Salesperson').apply(lambda x: x.nlargest(2, 'Revenue'))
print("\nTop 2 Transactions per Salesperson:\n", top_2_transactions_per_salesperson)
```

```
Top 2 Transactions per Salesperson:
                     Date Product Name  Units Sold  Revenue Region  \
Salesperson
Alice       11 2024-03-24     Widget B          14   847.99   East
            5  2024-02-11     Widget B          14   603.22   West
Bob         9  2024-03-10     Widget C           6   974.03  South
            12 2024-03-31     Widget B          17   937.64   West
Charlie     16 2024-04-28     Widget B          14   804.63  South
            6  2024-02-18     Widget B           6   742.65  South
David       0  2024-01-07     Widget B          14   809.91  North
            2  2024-01-21     Widget B          17   684.44  South

                Discount Offered (%) Salesperson  Revenue After Discount  \
Salesperson
Alice       11                 24.52       Alice              640.062852
            5                   6.90       Alice              561.597820
Bob         9                  25.85         Bob              722.243245
            12                  7.30         Bob              869.192280
Charlie     16                  2.94     Charlie              780.973878
            6                  11.88     Charlie              654.423180
David       0                   7.00       David              753.216300
            2                  21.18       David              539.475608

                Revenue Per Unit
Salesperson
Alice       11         60.570714
            5          43.087143
Bob         9         162.338333
            12         55.155294
Charlie     16         57.473571
            6         123.775000
David       0          57.850714
            2          40.261176
```

In [21]:

```python
# 16. Cumulative revenue for each product per day
cumulative_revenue_per_product = df.groupby(['Date', 'Product Name'])['Revenue'].sum().groupby(level=1).cumsum()
print("\nCumulative Revenue per Product per Day:\n", cumulative_revenue_per_product)
```

```
Cumulative Revenue per Product per Day:
 Date        Product Name
2024-01-07  Widget B         809.91
2024-01-14  Widget A         200.50
2024-01-21  Widget B        1494.35
2024-01-28  Widget A         526.19
2024-02-04  Widget B        2224.08
2024-02-11  Widget B        2827.30
2024-02-18  Widget B        3569.95
2024-02-25  Widget B        4180.40
2024-03-03  Widget C         346.44
2024-03-10  Widget C        1320.47
2024-03-17  Widget C        2246.20
2024-03-24  Widget B        5028.39
2024-03-31  Widget B        5966.03
2024-04-07  Widget B        6480.54
2024-04-14  Widget C        2601.45
2024-04-21  Widget B        6830.07
2024-04-28  Widget B        7634.70
2024-05-05  Widget B        7836.80
2024-05-12  Widget B        8315.01
2024-05-19  Widget C        2839.05
Name: Revenue, dtype: float64
```

In [22]:
```
# 17. Average revenue generated with discount vs without
avg_revenue_discount_vs_no_discount = df.groupby('Product Name').agg({
    'Revenue': ['mean'],
    'Discount Offered (%)': ['count']
}).rename(columns={('Revenue', 'mean'): 'Avg Revenue', ('Discount Offered (%)', 'count'): 'Transaction Count'})
print("\nAverage Revenue with vs without Discounts:\n", avg_revenue_discount_vs_no_discount)
```

```
Average Revenue with vs without Discounts:
                 Revenue Discount Offered (%)
                   mean                count
Product Name
Widget A      263.095000                    2
Widget B      639.616154                   13
Widget C      567.810000                    5
```

In [23]:
```
# 18. Weighted average discount offered by each salesperson
weighted_avg_discount = df.groupby('Salesperson').apply(
    lambda x: np.average(x['Discount Offered (%)'], weights=x['Revenue'])
)
print("\nWeighted Average Discount Offered by Each Salesperson:\n", weighted_avg_discount)
```

```
Weighted Average Discount Offered by Each Salesperson:
 Salesperson
Alice      13.158955
Bob        17.347127
Charlie     9.669257
David      12.359509
dtype: float64
```

In [24]:
```
# 19. Percentage of total revenue each region contributes
total_revenue = df['Revenue'].sum()
percentage_revenue_per_region = (revenue_per_region / total_revenue) * 100
print("\nPercentage of Total Revenue per Region:\n", percentage_revenue_per_region)
```

```
Percentage of Total Revenue per Region:
 Region
East     15.185634
North    17.346461
South    40.246313
West     27.221592
Name: Revenue, dtype: float64
```

In [ ]: