Homework Rakamin Trial Class - Mini Case

Mengukur Performa Penjualan Ritel Online

Load Data

```
import pandas as pd

data = pd.read_csv("C:\\Users\\mohra\\Downloads\\
online_retail_II.csv")
data
```

```
        Invoice StockCode                          Description
Quantity  \
0        489434     85048  15CM CHRISTMAS GLASS BALL 20 LIGHTS
12
1        489434    79323P                   PINK CHERRY LIGHTS
12
2        489434    79323W                  WHITE CHERRY LIGHTS
12
3        489434     22041         RECORD FRAME 7" SINGLE SIZE
48
4        489434     21232       STRAWBERRY CERAMIC TRINKET BOX
24
...         ...       ...                                ...       .
..
1067366  581587     22899         CHILDREN'S APRON DOLLY GIRL
6
1067367  581587     23254        CHILDRENS CUTLERY DOLLY GIRL
4
1067368  581587     23255      CHILDRENS CUTLERY CIRCUS PARADE
4
1067369  581587     22138        BAKING SET 9 PIECE RETROSPOT
3
1067370  581587      POST                              POSTAGE
1

                  InvoiceDate  Price  Customer ID         Country
0        2009-12-01 07:45:00   6.95      13085.0  United Kingdom
1        2009-12-01 07:45:00   6.75      13085.0  United Kingdom
2        2009-12-01 07:45:00   6.75      13085.0  United Kingdom
3        2009-12-01 07:45:00   2.10      13085.0  United Kingdom
4        2009-12-01 07:45:00   1.25      13085.0  United Kingdom
...                      ...    ...          ...             ...
1067366  2011-12-09 12:50:00   2.10      12680.0          France
1067367  2011-12-09 12:50:00   4.15      12680.0          France
1067368  2011-12-09 12:50:00   4.15      12680.0          France
```

```
1067369  2011-12-09 12:50:00   4.95      12680.0          France
1067370  2011-12-09 12:50:00  18.00      12680.0          France

[1067371 rows x 8 columns]
```

Section 1

Create New Feature: Year

```
data["InvoiceDate"] = pd.to_datetime(data["InvoiceDate"])
data["Year"] = data["InvoiceDate"].dt.year

data
```

```
        Invoice StockCode                          Description
Quantity  \
0        489434     85048  15CM CHRISTMAS GLASS BALL 20 LIGHTS
12
1        489434    79323P                   PINK CHERRY LIGHTS
12
2        489434    79323W                  WHITE CHERRY LIGHTS
12
3        489434     22041         RECORD FRAME 7" SINGLE SIZE
48
4        489434     21232       STRAWBERRY CERAMIC TRINKET BOX
24
...         ...       ...                                  ...        .
..
1067366  581587     22899         CHILDREN'S APRON DOLLY GIRL
6
1067367  581587     23254        CHILDRENS CUTLERY DOLLY GIRL
4
1067368  581587     23255      CHILDRENS CUTLERY CIRCUS PARADE
4
1067369  581587     22138        BAKING SET 9 PIECE RETROSPOT
3
1067370  581587      POST                              POSTAGE
1

                InvoiceDate  Price  Customer ID         Country  Year

0       2009-12-01 07:45:00   6.95      13085.0  United Kingdom  2009

1       2009-12-01 07:45:00   6.75      13085.0  United Kingdom  2009

2       2009-12-01 07:45:00   6.75      13085.0  United Kingdom  2009

3       2009-12-01 07:45:00   2.10      13085.0  United Kingdom  2009

4       2009-12-01 07:45:00   1.25      13085.0  United Kingdom  2009
```

```
...                     ...    ...          ...             ...   ...
1067366 2011-12-09 12:50:00   2.10      12680.0          France  2011
1067367 2011-12-09 12:50:00   4.15      12680.0          France  2011
1067368 2011-12-09 12:50:00   4.15      12680.0          France  2011
1067369 2011-12-09 12:50:00   4.95      12680.0          France  2011
1067370 2011-12-09 12:50:00  18.00      12680.0          France  2011

[1067371 rows x 9 columns]
```

Filtering Data

```
sales = data[(data["Quantity"] > 0) &
(~data["Invoice"].str.startswith("C"))]

sales
```

```
        Invoice StockCode                          Description
Quantity  \
0        489434     85048  15CM CHRISTMAS GLASS BALL 20 LIGHTS
12
1        489434    79323P                   PINK CHERRY LIGHTS
12
2        489434    79323W                  WHITE CHERRY LIGHTS
12
3        489434     22041         RECORD FRAME 7" SINGLE SIZE
48
4        489434     21232       STRAWBERRY CERAMIC TRINKET BOX
24
...         ...       ...                                  ...       .
..
1067366  581587     22899         CHILDREN'S APRON DOLLY GIRL
6
1067367  581587     23254        CHILDRENS CUTLERY DOLLY GIRL
4
1067368  581587     23255      CHILDRENS CUTLERY CIRCUS PARADE
4
1067369  581587     22138        BAKING SET 9 PIECE RETROSPOT
3
1067370  581587      POST                              POSTAGE
1

                InvoiceDate  Price  Customer ID         Country  Year

0       2009-12-01 07:45:00   6.95      13085.0  United Kingdom  2009
```

```
1       2009-12-01 07:45:00   6.75      13085.0  United Kingdom  2009
2       2009-12-01 07:45:00   6.75      13085.0  United Kingdom  2009
3       2009-12-01 07:45:00   2.10      13085.0  United Kingdom  2009
4       2009-12-01 07:45:00   1.25      13085.0  United Kingdom  2009
...                     ...    ...          ...             ...   ...
1067366 2011-12-09 12:50:00   2.10      12680.0          France  2011
1067367 2011-12-09 12:50:00   4.15      12680.0          France  2011
1067368 2011-12-09 12:50:00   4.15      12680.0          France  2011
1067369 2011-12-09 12:50:00   4.95      12680.0          France  2011
1067370 2011-12-09 12:50:00  18.00      12680.0          France  2011

[1044420 rows x 9 columns]
```

Create New Feature: Revenue

```
sales["Revenue"] = sales["Quantity"] * sales["Price"]

sales
```

```
C:\Users\mohra\AppData\Local\Temp\ipykernel_20548\605054434.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
  sales["Revenue"] = sales["Quantity"] * sales["Price"]
```

```
        Invoice StockCode                          Description
Quantity  \
0        489434     85048  15CM CHRISTMAS GLASS BALL 20 LIGHTS
12
1        489434    79323P                   PINK CHERRY LIGHTS
12
2        489434    79323W                  WHITE CHERRY LIGHTS
12
3        489434     22041         RECORD FRAME 7" SINGLE SIZE
48
4        489434     21232       STRAWBERRY CERAMIC TRINKET BOX
```

```
24
...         ...       ...                                  ...       .
..
1067366  581587     22899         CHILDREN'S APRON DOLLY GIRL
6
1067367  581587     23254        CHILDRENS CUTLERY DOLLY GIRL
4
1067368  581587     23255      CHILDRENS CUTLERY CIRCUS PARADE
4
1067369  581587     22138        BAKING SET 9 PIECE RETROSPOT
3
1067370  581587      POST                              POSTAGE
1

                 InvoiceDate  Price  Customer ID         Country  Year
Revenue
0       2009-12-01 07:45:00   6.95      13085.0  United Kingdom  2009
83.40
1       2009-12-01 07:45:00   6.75      13085.0  United Kingdom  2009
81.00
2       2009-12-01 07:45:00   6.75      13085.0  United Kingdom  2009
81.00
3       2009-12-01 07:45:00   2.10      13085.0  United Kingdom  2009
100.80
4       2009-12-01 07:45:00   1.25      13085.0  United Kingdom  2009
30.00
...                     ...    ...          ...             ...   ...
...
1067366 2011-12-09 12:50:00   2.10      12680.0          France  2011
12.60
1067367 2011-12-09 12:50:00   4.15      12680.0          France  2011
16.60
1067368 2011-12-09 12:50:00   4.15      12680.0          France  2011
16.60
1067369 2011-12-09 12:50:00   4.95      12680.0          France  2011
14.85
1067370 2011-12-09 12:50:00  18.00      12680.0          France  2011
18.00

[1044420 rows x 10 columns]
```

Average of Revenue per Year

```
average_per_year = sales.groupby("Year")["Revenue"].mean()

average_per_year

Year
2009    18.713273
2010    19.908987
```

```
2011    20.058767
Name: Revenue, dtype: float64
```

Interpretation

```
# Interpretation of Section 1
from IPython.display import Markdown

# Calculate the average revenue per year
average_per_year = sales.groupby("Year")["Revenue"].mean()

# Generate Markdown interpretation
interpretation = "### Interpretation of Section 1\n\n"
interpretation += "The table below shows the average annual revenue
for each year:\n\n"
interpretation += "| Year | Average Revenue |\n"
interpretation += "|------|-----------------|\n"

for year, average_revenue in average_revenue_per_year.items():
    interpretation += f"| {year} | {average_revenue:.2f} |\n"

# Display the interpretation in Markdown format
Markdown(interpretation)
```

```
<IPython.core.display.Markdown object>
```

Section 2

Filtering Data

Customers who finished their purchases

```
finished = sales.dropna(subset=["Customer ID"])

finished
```

```
        Invoice StockCode                          Description
Quantity  \
0        489434     85048  15CM CHRISTMAS GLASS BALL 20 LIGHTS
12
1        489434    79323P                   PINK CHERRY LIGHTS
12
2        489434    79323W                  WHITE CHERRY LIGHTS
12
3        489434     22041         RECORD FRAME 7" SINGLE SIZE
48
4        489434     21232       STRAWBERRY CERAMIC TRINKET BOX
24
...         ...       ...                                 ...       .
..
```

```
1067366  581587     22899         CHILDREN'S APRON DOLLY GIRL
6
1067367  581587     23254        CHILDRENS CUTLERY DOLLY GIRL
4
1067368  581587     23255      CHILDRENS CUTLERY CIRCUS PARADE
4
1067369  581587     22138        BAKING SET 9 PIECE RETROSPOT
3
1067370  581587      POST                              POSTAGE
1

                 InvoiceDate  Price  Customer ID         Country  Year
Revenue
0       2009-12-01 07:45:00   6.95      13085.0  United Kingdom  2009
83.40
1       2009-12-01 07:45:00   6.75      13085.0  United Kingdom  2009
81.00
2       2009-12-01 07:45:00   6.75      13085.0  United Kingdom  2009
81.00
3       2009-12-01 07:45:00   2.10      13085.0  United Kingdom  2009
100.80
4       2009-12-01 07:45:00   1.25      13085.0  United Kingdom  2009
30.00
...                     ...    ...          ...             ...   ...
...
1067366 2011-12-09 12:50:00   2.10      12680.0          France  2011
12.60
1067367 2011-12-09 12:50:00   4.15      12680.0          France  2011
16.60
1067368 2011-12-09 12:50:00   4.15      12680.0          France  2011
16.60
1067369 2011-12-09 12:50:00   4.95      12680.0          France  2011
14.85
1067370 2011-12-09 12:50:00  18.00      12680.0          France  2011
18.00

[805620 rows x 10 columns]
```

Customers who canceled their purchases

```
canceled = sales[sales['Invoice'].str.startswith('C')]

canceled

Empty DataFrame
Columns: [Invoice, StockCode, Description, Quantity, InvoiceDate,
Price, Customer ID, Country, Year, Revenue]
Index: []
```

Number of Finished and Canceled Transactions Each Year

```
finished_transaction = finished.groupby('Year').size()
canceled_transaction = canceled.groupby('Year').size()

finished_transaction
```

```
Year
2009     30761
2010    403094
2011    371765
dtype: int64
```

```
canceled_transaction
```

```
Series([], dtype: int64)
```

Cancellation Rate

```
canceled_rate = (canceled_transaction / (canceled_transaction +
finished_transaction) * 100)

canceled_rate
```

```
Year
2009   NaN
2010   NaN
2011   NaN
dtype: float64
```

Interpretation

```
# Interpretation of Section 2
from IPython.display import Markdown

# Customers who finished their purchases
finished = sales.dropna(subset=["Customer ID"])

# Customers who canceled their purchases
canceled = sales[sales['Invoice'].str.startswith('C')]

# Number of Finished and Canceled Transactions Each Year
finished_transaction = finished.groupby('Year').size()
canceled_transaction = canceled.groupby('Year').size()

# Cancellation Rate
canceled_rate = (canceled_transaction / (canceled_transaction +
finished_transaction) * 100)

# Generate Markdown interpretation
interpretation_section2 = "### Interpretation of Section 2\n\n"
```

```
interpretation_section2 += "In Section 2, we analyze the transactions
of customers who either finished their purchases or canceled them.\n\
n"
interpretation_section2 += f"There were {finished_transaction.sum()}
unique customers who successfully finished their transactions, "
interpretation_section2 += f"and {canceled_transaction.sum()} unique
customers who canceled their purchases during the analyzed period.\n\
n"
interpretation_section2 += "The total number of finished transactions
and canceled transactions each year are as follows:\n\n"
interpretation_section2 += "| Year | Finished Transactions | Canceled
Transactions |\n"
interpretation_section2 +=
"|------|-----------------------|-----------------------|\n"

for year in finished_transaction.index:
    interpretation_section2 += f"| {year} |
{finished_transaction[year]} | {canceled_transaction.get(year, 0)} |\
n"

interpretation_section2 += f"\nThe overall cancellation rate, which
represents the percentage of customers who canceled their "
interpretation_section2 += f"orders out of all customers, is
{canceled_rate.values[0]:.2f}%."

# Display the interpretation in Markdown format
Markdown(interpretation_section2)
```

```
<IPython.core.display.Markdown object>
```