```
pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\mbdev\anaconda3\lib\
site-packages (2.2.2)
Requirement already satisfied: numpy>=1.26.0 in c:\users\mbdev\
anaconda3\lib\site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\
mbdev\anaconda3\lib\site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\mbdev\
anaconda3\lib\site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\mbdev\
anaconda3\lib\site-packages (from pandas) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\mbdev\anaconda3\
lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

# import the data set
train_df = pd.read_csv(r"C:\Users\mbdev\Desktop\train_data.csv\
train_data.csv")
test_df = pd.read_csv(r"C:\Users\mbdev\Desktop\train_data.csv\
test_data.csv")
submission_df = pd.read_csv(r"C:\Users\mbdev\Desktop\train_data.csv\
sample_submission.csv")

# Step 1: Preview the basic checks...

print(train_df.head())
```

```
         date  product_identifier  department_identifier  \
0  2012-01-01                  74                     11
1  2012-01-01                 337                     11
2  2012-01-01                 423                     12
3  2012-01-01                 432                     12
4  2012-01-01                 581                     21

          category_of_product  outlet        state  sales
0                      others     111  Maharashtra      0
1                      others     111  Maharashtra      1
2                      others     111  Maharashtra      0
```

```
3                     others     111  Maharashtra      0
4  fast_moving_consumer_goods     111  Maharashtra      0

print(test_df.head())

   id        date  product_identifier  department_identifier  \
0   1  2014-03-01                  74                     11
1   2  2014-03-01                 337                     11
2   3  2014-03-01                 423                     12
3   4  2014-03-01                 432                     12
4   5  2014-03-01                 581                     21

          category_of_product  outlet       state
0                      others     111  Maharashtra
1                      others     111  Maharashtra
2                      others     111  Maharashtra
3                      others     111  Maharashtra
4  fast_moving_consumer_goods     111  Maharashtra

print(train_df.tail())

              date  product_identifier  department_identifier  \
394995  2014-02-28                2932                     33
394996  2014-02-28                2935                     33
394997  2014-02-28                3004                     33
394998  2014-02-28                3008                     33
394999  2014-02-28                3021                     33

       category_of_product  outlet   state  sales
394995     drinks_and_food     333  Kerala      2
394996     drinks_and_food     333  Kerala      8
394997     drinks_and_food     333  Kerala      0
394998     drinks_and_food     333  Kerala      0
394999     drinks_and_food     333  Kerala      0

print(submission_df.head())

   id  sales
0   1      0
1   2      0
2   3      0
3   4      0
4   5      0

print(test_df.tail())

          id        date  product_identifier  department_identifier  \
15495  15496  2014-03-31                2932                     33
15496  15497  2014-03-31                2935                     33
15497  15498  2014-03-31                3004                     33
15498  15499  2014-03-31                3008                     33
```

```
15499  15500  2014-03-31                3021                      33

      category_of_product  outlet   state
15495     drinks_and_food     333  Kerala
15496     drinks_and_food     333  Kerala
15497     drinks_and_food     333  Kerala
15498     drinks_and_food     333  Kerala
15499     drinks_and_food     333  Kerala

print(submission_df.tail())

          id  sales
15495  15496      0
15496  15497      0
15497  15498      0
15498  15499      0
15499  15500      0

# to print the shape (To see how many rows and columns each DataFrame)

print("Train data shape:", train_df.shape)
print("Test data shape:", test_df.shape)
print("Submission file shape:", submission_df.shape)

Train data shape: (395000, 7)
Test data shape: (15500, 7)
Submission file shape: (15500, 2)

# to print the size(total number of elements (rows × columns))

print("Train data size:", train_df.size)
print("Test data size:", test_df.size)
print("Submission file size:", submission_df.size)

Train data size: 2765000
Test data size: 108500
Submission file size: 31000

# to print the info(structure of your dataset)

print("Train data :", train_df.info)

Train data : <bound method DataFrame.info of               date
product_identifier  department_identifier  \
0       2012-01-01                  74                      11
1       2012-01-01                 337                      11
2       2012-01-01                 423                      12
3       2012-01-01                 432                      12
4       2012-01-01                 581                      21
...            ...                 ...                     ...
394995  2014-02-28                2932                      33
```

```
394996  2014-02-28                2935                      33
394997  2014-02-28                3004                      33
394998  2014-02-28                3008                      33
394999  2014-02-28                3021                      33

               category_of_product  outlet        state  sales
0                           others     111  Maharashtra      0
1                           others     111  Maharashtra      1
2                           others     111  Maharashtra      0
3                           others     111  Maharashtra      0
4       fast_moving_consumer_goods     111  Maharashtra      0
...                            ...     ...          ...    ...
394995              drinks_and_food     333       Kerala      2
394996              drinks_and_food     333       Kerala      8
394997              drinks_and_food     333       Kerala      0
394998              drinks_and_food     333       Kerala      0
394999              drinks_and_food     333       Kerala      0

[395000 rows x 7 columns]>
```

```
print("Test data :", test_df.info)
```

```
Test data : <bound method DataFrame.info of           id        date
product_identifier  department_identifier  \
0          1  2014-03-01                  74                      11
1          2  2014-03-01                 337                      11
2          3  2014-03-01                 423                      12
3          4  2014-03-01                 432                      12
4          5  2014-03-01                 581                      21
...      ...         ...                 ...                     ...
15495  15496  2014-03-31                2932                      33
15496  15497  2014-03-31                2935                      33
15497  15498  2014-03-31                3004                      33
15498  15499  2014-03-31                3008                      33
15499  15500  2014-03-31                3021                      33

               category_of_product  outlet        state
0                           others     111  Maharashtra
1                           others     111  Maharashtra
2                           others     111  Maharashtra
3                           others     111  Maharashtra
4       fast_moving_consumer_goods     111  Maharashtra
...                            ...     ...          ...
15495               drinks_and_food     333       Kerala
15496               drinks_and_food     333       Kerala
15497               drinks_and_food     333       Kerala
15498               drinks_and_food     333       Kerala
15499               drinks_and_food     333       Kerala

[15500 rows x 7 columns]>
```

```
print("Submission file :", submission_df.info)
```

```
Submission file : <bound method DataFrame.info of           id  sales
0          1      0
1          2      0
2          3      0
3          4      0
4          5      0
...      ...    ...
15495  15496      0
15496  15497      0
15497  15498      0
15498  15499      0
15499  15500      0

[15500 rows x 2 columns]>
```

```
# to print the describe

print("Train data size:", train_df.describe())
```

```
Train data size:         product_identifier  department_identifier
outlet          sales
count       395000.000000          395000.000000  395000.000000
395000.000000
mean          1509.960000              24.460000     211.200000
1.228919
std            809.799518               6.337863      91.161291
3.595266
min             74.000000              11.000000     111.000000
0.000000
25%            926.000000              21.000000     113.000000
0.000000
50%           1325.000000              22.000000     221.500000
0.000000
75%           1753.000000              31.000000     331.000000
1.000000
max           3021.000000              33.000000     333.000000
293.000000
```

```
print("Test data size:", test_df.describe())
```

```
Test data size:                   id  product_identifier
department_identifier        outlet
count  15500.000000        15500.000000           15500.000000
15500.000000
mean    7750.500000         1509.960000              24.460000
211.200000
std     4474.608921          809.824616               6.338059
91.164117
min        1.000000           74.000000              11.000000
```

```
111.000000
25%     3875.750000          926.000000              21.000000
113.000000
50%     7750.500000         1325.000000              22.000000
221.500000
75%    11625.250000         1753.000000              31.000000
331.000000
max    15500.000000         3021.000000              33.000000
333.000000
```

```
print("Submission file size:", submission_df.describe())
```

```
Submission file size:                   id    sales
count  15500.000000  15500.0
mean    7750.500000      0.0
std     4474.608921      0.0
min        1.000000      0.0
25%     3875.750000      0.0
50%     7750.500000      0.0
75%    11625.250000      0.0
max    15500.000000      0.0
```

```
# to print the type of the data used in the dataset
```

```
print("Train data :", train_df.dtypes)
```

```
Train data : date                       object
product_identifier        int64
department_identifier     int64
category_of_product      object
outlet                    int64
state                    object
sales                     int64
dtype: object
```

```
print("Test data :", test_df.dtypes)
```

```
Test data : id                          int64
date                     object
product_identifier        int64
department_identifier     int64
category_of_product      object
outlet                    int64
state                    object
dtype: object
```

```
print("Submission file :", submission_df.dtypes)
```

```
Submission file : id       int64
sales    int64
dtype: object
```

```
train_df['source'] = 'train'
test_df['source'] = 'test'
test_df['target'] = None  # Fill with None or NaN
combined_df = pd.concat([train_df, test_df], ignore_index=True)

print(combined_df['source'].value_counts())
print(combined_df.size)
```

```
source
train    395000
test      15500
Name: count, dtype: int64
4105000
```

```
combined_df.head
```

```
<bound method NDFrame.head of                date  product_identifier
department_identifier  \
0       2012-01-01                  74                     11
1       2012-01-01                 337                     11
2       2012-01-01                 423                     12
3       2012-01-01                 432                     12
4       2012-01-01                 581                     21
...            ...                 ...                    ...
410495  2014-03-31                2932                     33
410496  2014-03-31                2935                     33
410497  2014-03-31                3004                     33
410498  2014-03-31                3008                     33
410499  2014-03-31                3021                     33

               category_of_product  outlet        state  sales source
\
0                           others     111  Maharashtra    0.0  train

1                           others     111  Maharashtra    1.0  train

2                           others     111  Maharashtra    0.0  train

3                           others     111  Maharashtra    0.0  train

4       fast_moving_consumer_goods     111  Maharashtra    0.0  train

...                            ...     ...          ...    ...    ...

410495             drinks_and_food     333       Kerala    NaN   test

410496             drinks_and_food     333       Kerala    NaN   test

410497             drinks_and_food     333       Kerala    NaN   test

410498             drinks_and_food     333       Kerala    NaN   test
```

```
410499             drinks_and_food     333       Kerala    NaN   test


              id target
0            NaN    NaN
1            NaN    NaN
2            NaN    NaN
3            NaN    NaN
4            NaN    NaN
...          ...    ...
410495  15496.0   None
410496  15497.0   None
410497  15498.0   None
410498  15499.0   None
410499  15500.0   None

[410500 rows x 10 columns]>
```

```
print("combined data:", combined_df.describe())
```

```
combined data:        product_identifier  department_identifier
outlet  \
count       410500.000000          410500.000000  410500.000000
mean          1509.960000              24.460000     211.200000
std            809.799479               6.337862      91.161287
min             74.000000              11.000000     111.000000
25%            926.000000              21.000000     113.000000
50%           1325.000000              22.000000     221.500000
75%           1753.000000              31.000000     331.000000
max           3021.000000              33.000000     333.000000

                sales            id
count  395000.000000  15500.000000
mean        1.228919   7750.500000
std         3.595266   4474.608921
min         0.000000      1.000000
25%         0.000000   3875.750000
50%         0.000000   7750.500000
75%         1.000000  11625.250000
max       293.000000  15500.000000
```

```
# Get unique values for each column in the DataFrame
for col in combined_df.columns:
    print(f"Unique values in {col}:\n", combined_df[col].unique(), "\
n")
```

```
Unique values in date:
 ['2012-01-01' '2012-01-02' '2012-01-03' '2012-01-04' '2012-01-05'
 '2012-01-06' '2012-01-07' '2012-01-08' '2012-01-09' '2012-01-10'
 '2012-01-11' '2012-01-12' '2012-01-13' '2012-01-14' '2012-01-15'
```

```
'2012-01-16' '2012-01-17' '2012-01-18' '2012-01-19' '2012-01-20'
'2012-01-21' '2012-01-22' '2012-01-23' '2012-01-24' '2012-01-25'
'2012-01-26' '2012-01-27' '2012-01-28' '2012-01-29' '2012-01-30'
'2012-01-31' '2012-02-01' '2012-02-02' '2012-02-03' '2012-02-04'
'2012-02-05' '2012-02-06' '2012-02-07' '2012-02-08' '2012-02-09'
'2012-02-10' '2012-02-11' '2012-02-12' '2012-02-13' '2012-02-14'
'2012-02-15' '2012-02-16' '2012-02-17' '2012-02-18' '2012-02-19'
'2012-02-20' '2012-02-21' '2012-02-22' '2012-02-23' '2012-02-24'
'2012-02-25' '2012-02-26' '2012-02-27' '2012-02-28' '2012-02-29'
'2012-03-01' '2012-03-02' '2012-03-03' '2012-03-04' '2012-03-05'
'2012-03-06' '2012-03-07' '2012-03-08' '2012-03-09' '2012-03-10'
'2012-03-11' '2012-03-12' '2012-03-13' '2012-03-14' '2012-03-15'
'2012-03-16' '2012-03-17' '2012-03-18' '2012-03-19' '2012-03-20'
'2012-03-21' '2012-03-22' '2012-03-23' '2012-03-24' '2012-03-25'
'2012-03-26' '2012-03-27' '2012-03-28' '2012-03-29' '2012-03-30'
'2012-03-31' '2012-04-01' '2012-04-02' '2012-04-03' '2012-04-04'
'2012-04-05' '2012-04-06' '2012-04-07' '2012-04-08' '2012-04-09'
'2012-04-10' '2012-04-11' '2012-04-12' '2012-04-13' '2012-04-14'
'2012-04-15' '2012-04-16' '2012-04-17' '2012-04-18' '2012-04-19'
'2012-04-20' '2012-04-21' '2012-04-22' '2012-04-23' '2012-04-24'
'2012-04-25' '2012-04-26' '2012-04-27' '2012-04-28' '2012-04-29'
'2012-04-30' '2012-05-01' '2012-05-02' '2012-05-03' '2012-05-04'
'2012-05-05' '2012-05-06' '2012-05-07' '2012-05-08' '2012-05-09'
'2012-05-10' '2012-05-11' '2012-05-12' '2012-05-13' '2012-05-14'
'2012-05-15' '2012-05-16' '2012-05-17' '2012-05-18' '2012-05-19'
'2012-05-20' '2012-05-21' '2012-05-22' '2012-05-23' '2012-05-24'
'2012-05-25' '2012-05-26' '2012-05-27' '2012-05-28' '2012-05-29'
'2012-05-30' '2012-05-31' '2012-06-01' '2012-06-02' '2012-06-03'
'2012-06-04' '2012-06-05' '2012-06-06' '2012-06-07' '2012-06-08'
'2012-06-09' '2012-06-10' '2012-06-11' '2012-06-12' '2012-06-13'
'2012-06-14' '2012-06-15' '2012-06-16' '2012-06-17' '2012-06-18'
'2012-06-19' '2012-06-20' '2012-06-21' '2012-06-22' '2012-06-23'
'2012-06-24' '2012-06-25' '2012-06-26' '2012-06-27' '2012-06-28'
'2012-06-29' '2012-06-30' '2012-07-01' '2012-07-02' '2012-07-03'
'2012-07-04' '2012-07-05' '2012-07-06' '2012-07-07' '2012-07-08'
'2012-07-09' '2012-07-10' '2012-07-11' '2012-07-12' '2012-07-13'
'2012-07-14' '2012-07-15' '2012-07-16' '2012-07-17' '2012-07-18'
'2012-07-19' '2012-07-20' '2012-07-21' '2012-07-22' '2012-07-23'
'2012-07-24' '2012-07-25' '2012-07-26' '2012-07-27' '2012-07-28'
'2012-07-29' '2012-07-30' '2012-07-31' '2012-08-01' '2012-08-02'
'2012-08-03' '2012-08-04' '2012-08-05' '2012-08-06' '2012-08-07'
'2012-08-08' '2012-08-09' '2012-08-10' '2012-08-11' '2012-08-12'
'2012-08-13' '2012-08-14' '2012-08-15' '2012-08-16' '2012-08-17'
'2012-08-18' '2012-08-19' '2012-08-20' '2012-08-21' '2012-08-22'
'2012-08-23' '2012-08-24' '2012-08-25' '2012-08-26' '2012-08-27'
'2012-08-28' '2012-08-29' '2012-08-30' '2012-08-31' '2012-09-01'
'2012-09-02' '2012-09-03' '2012-09-04' '2012-09-05' '2012-09-06'
'2012-09-07' '2012-09-08' '2012-09-09' '2012-09-10' '2012-09-11'
'2012-09-12' '2012-09-13' '2012-09-14' '2012-09-15' '2012-09-16'
```

```
'2012-09-17' '2012-09-18' '2012-09-19' '2012-09-20' '2012-09-21'
'2012-09-22' '2012-09-23' '2012-09-24' '2012-09-25' '2012-09-26'
'2012-09-27' '2012-09-28' '2012-09-29' '2012-09-30' '2012-10-01'
'2012-10-02' '2012-10-03' '2012-10-04' '2012-10-05' '2012-10-06'
'2012-10-07' '2012-10-08' '2012-10-09' '2012-10-10' '2012-10-11'
'2012-10-12' '2012-10-13' '2012-10-14' '2012-10-15' '2012-10-16'
'2012-10-17' '2012-10-18' '2012-10-19' '2012-10-20' '2012-10-21'
'2012-10-22' '2012-10-23' '2012-10-24' '2012-10-25' '2012-10-26'
'2012-10-27' '2012-10-28' '2012-10-29' '2012-10-30' '2012-10-31'
'2012-11-01' '2012-11-02' '2012-11-03' '2012-11-04' '2012-11-05'
'2012-11-06' '2012-11-07' '2012-11-08' '2012-11-09' '2012-11-10'
'2012-11-11' '2012-11-12' '2012-11-13' '2012-11-14' '2012-11-15'
'2012-11-16' '2012-11-17' '2012-11-18' '2012-11-19' '2012-11-20'
'2012-11-21' '2012-11-22' '2012-11-23' '2012-11-24' '2012-11-25'
'2012-11-26' '2012-11-27' '2012-11-28' '2012-11-29' '2012-11-30'
'2012-12-01' '2012-12-02' '2012-12-03' '2012-12-04' '2012-12-05'
'2012-12-06' '2012-12-07' '2012-12-08' '2012-12-09' '2012-12-10'
'2012-12-11' '2012-12-12' '2012-12-13' '2012-12-14' '2012-12-15'
'2012-12-16' '2012-12-17' '2012-12-18' '2012-12-19' '2012-12-20'
'2012-12-21' '2012-12-22' '2012-12-23' '2012-12-24' '2012-12-25'
'2012-12-26' '2012-12-27' '2012-12-28' '2012-12-29' '2012-12-30'
'2012-12-31' '2013-01-01' '2013-01-02' '2013-01-03' '2013-01-04'
'2013-01-05' '2013-01-06' '2013-01-07' '2013-01-08' '2013-01-09'
'2013-01-10' '2013-01-11' '2013-01-12' '2013-01-13' '2013-01-14'
'2013-01-15' '2013-01-16' '2013-01-17' '2013-01-18' '2013-01-19'
'2013-01-20' '2013-01-21' '2013-01-22' '2013-01-23' '2013-01-24'
'2013-01-25' '2013-01-26' '2013-01-27' '2013-01-28' '2013-01-29'
'2013-01-30' '2013-01-31' '2013-02-01' '2013-02-02' '2013-02-03'
'2013-02-04' '2013-02-05' '2013-02-06' '2013-02-07' '2013-02-08'
'2013-02-09' '2013-02-10' '2013-02-11' '2013-02-12' '2013-02-13'
'2013-02-14' '2013-02-15' '2013-02-16' '2013-02-17' '2013-02-18'
'2013-02-19' '2013-02-20' '2013-02-21' '2013-02-22' '2013-02-23'
'2013-02-24' '2013-02-25' '2013-02-26' '2013-02-27' '2013-02-28'
'2013-03-01' '2013-03-02' '2013-03-03' '2013-03-04' '2013-03-05'
'2013-03-06' '2013-03-07' '2013-03-08' '2013-03-09' '2013-03-10'
'2013-03-11' '2013-03-12' '2013-03-13' '2013-03-14' '2013-03-15'
'2013-03-16' '2013-03-17' '2013-03-18' '2013-03-19' '2013-03-20'
'2013-03-21' '2013-03-22' '2013-03-23' '2013-03-24' '2013-03-25'
'2013-03-26' '2013-03-27' '2013-03-28' '2013-03-29' '2013-03-30'
'2013-03-31' '2013-04-01' '2013-04-02' '2013-04-03' '2013-04-04'
'2013-04-05' '2013-04-06' '2013-04-07' '2013-04-08' '2013-04-09'
'2013-04-10' '2013-04-11' '2013-04-12' '2013-04-13' '2013-04-14'
'2013-04-15' '2013-04-16' '2013-04-17' '2013-04-18' '2013-04-19'
'2013-04-20' '2013-04-21' '2013-04-22' '2013-04-23' '2013-04-24'
'2013-04-25' '2013-04-26' '2013-04-27' '2013-04-28' '2013-04-29'
'2013-04-30' '2013-05-01' '2013-05-02' '2013-05-03' '2013-05-04'
'2013-05-05' '2013-05-06' '2013-05-07' '2013-05-08' '2013-05-09'
'2013-05-10' '2013-05-11' '2013-05-12' '2013-05-13' '2013-05-14'
'2013-05-15' '2013-05-16' '2013-05-17' '2013-05-18' '2013-05-19'
```

```
'2013-05-20' '2013-05-21' '2013-05-22' '2013-05-23' '2013-05-24'
'2013-05-25' '2013-05-26' '2013-05-27' '2013-05-28' '2013-05-29'
'2013-05-30' '2013-05-31' '2013-06-01' '2013-06-02' '2013-06-03'
'2013-06-04' '2013-06-05' '2013-06-06' '2013-06-07' '2013-06-08'
'2013-06-09' '2013-06-10' '2013-06-11' '2013-06-12' '2013-06-13'
'2013-06-14' '2013-06-15' '2013-06-16' '2013-06-17' '2013-06-18'
'2013-06-19' '2013-06-20' '2013-06-21' '2013-06-22' '2013-06-23'
'2013-06-24' '2013-06-25' '2013-06-26' '2013-06-27' '2013-06-28'
'2013-06-29' '2013-06-30' '2013-07-01' '2013-07-02' '2013-07-03'
'2013-07-04' '2013-07-05' '2013-07-06' '2013-07-07' '2013-07-08'
'2013-07-09' '2013-07-10' '2013-07-11' '2013-07-12' '2013-07-13'
'2013-07-14' '2013-07-15' '2013-07-16' '2013-07-17' '2013-07-18'
'2013-07-19' '2013-07-20' '2013-07-21' '2013-07-22' '2013-07-23'
'2013-07-24' '2013-07-25' '2013-07-26' '2013-07-27' '2013-07-28'
'2013-07-29' '2013-07-30' '2013-07-31' '2013-08-01' '2013-08-02'
'2013-08-03' '2013-08-04' '2013-08-05' '2013-08-06' '2013-08-07'
'2013-08-08' '2013-08-09' '2013-08-10' '2013-08-11' '2013-08-12'
'2013-08-13' '2013-08-14' '2013-08-15' '2013-08-16' '2013-08-17'
'2013-08-18' '2013-08-19' '2013-08-20' '2013-08-21' '2013-08-22'
'2013-08-23' '2013-08-24' '2013-08-25' '2013-08-26' '2013-08-27'
'2013-08-28' '2013-08-29' '2013-08-30' '2013-08-31' '2013-09-01'
'2013-09-02' '2013-09-03' '2013-09-04' '2013-09-05' '2013-09-06'
'2013-09-07' '2013-09-08' '2013-09-09' '2013-09-10' '2013-09-11'
'2013-09-12' '2013-09-13' '2013-09-14' '2013-09-15' '2013-09-16'
'2013-09-17' '2013-09-18' '2013-09-19' '2013-09-20' '2013-09-21'
'2013-09-22' '2013-09-23' '2013-09-24' '2013-09-25' '2013-09-26'
'2013-09-27' '2013-09-28' '2013-09-29' '2013-09-30' '2013-10-01'
'2013-10-02' '2013-10-03' '2013-10-04' '2013-10-05' '2013-10-06'
'2013-10-07' '2013-10-08' '2013-10-09' '2013-10-10' '2013-10-11'
'2013-10-12' '2013-10-13' '2013-10-14' '2013-10-15' '2013-10-16'
'2013-10-17' '2013-10-18' '2013-10-19' '2013-10-20' '2013-10-21'
'2013-10-22' '2013-10-23' '2013-10-24' '2013-10-25' '2013-10-26'
'2013-10-27' '2013-10-28' '2013-10-29' '2013-10-30' '2013-10-31'
'2013-11-01' '2013-11-02' '2013-11-03' '2013-11-04' '2013-11-05'
'2013-11-06' '2013-11-07' '2013-11-08' '2013-11-09' '2013-11-10'
'2013-11-11' '2013-11-12' '2013-11-13' '2013-11-14' '2013-11-15'
'2013-11-16' '2013-11-17' '2013-11-18' '2013-11-19' '2013-11-20'
'2013-11-21' '2013-11-22' '2013-11-23' '2013-11-24' '2013-11-25'
'2013-11-26' '2013-11-27' '2013-11-28' '2013-11-29' '2013-11-30'
'2013-12-01' '2013-12-02' '2013-12-03' '2013-12-04' '2013-12-05'
'2013-12-06' '2013-12-07' '2013-12-08' '2013-12-09' '2013-12-10'
'2013-12-11' '2013-12-12' '2013-12-13' '2013-12-14' '2013-12-15'
'2013-12-16' '2013-12-17' '2013-12-18' '2013-12-19' '2013-12-20'
'2013-12-21' '2013-12-22' '2013-12-23' '2013-12-24' '2013-12-25'
'2013-12-26' '2013-12-27' '2013-12-28' '2013-12-29' '2013-12-30'
'2013-12-31' '2014-01-01' '2014-01-02' '2014-01-03' '2014-01-04'
'2014-01-05' '2014-01-06' '2014-01-07' '2014-01-08' '2014-01-09'
'2014-01-10' '2014-01-11' '2014-01-12' '2014-01-13' '2014-01-14'
'2014-01-15' '2014-01-16' '2014-01-17' '2014-01-18' '2014-01-19'
```

```
 '2014-01-20' '2014-01-21' '2014-01-22' '2014-01-23' '2014-01-24'
 '2014-01-25' '2014-01-26' '2014-01-27' '2014-01-28' '2014-01-29'
 '2014-01-30' '2014-01-31' '2014-02-01' '2014-02-02' '2014-02-03'
 '2014-02-04' '2014-02-05' '2014-02-06' '2014-02-07' '2014-02-08'
 '2014-02-09' '2014-02-10' '2014-02-11' '2014-02-12' '2014-02-13'
 '2014-02-14' '2014-02-15' '2014-02-16' '2014-02-17' '2014-02-18'
 '2014-02-19' '2014-02-20' '2014-02-21' '2014-02-22' '2014-02-23'
 '2014-02-24' '2014-02-25' '2014-02-26' '2014-02-27' '2014-02-28'
 '2014-03-01' '2014-03-02' '2014-03-03' '2014-03-04' '2014-03-05'
 '2014-03-06' '2014-03-07' '2014-03-08' '2014-03-09' '2014-03-10'
 '2014-03-11' '2014-03-12' '2014-03-13' '2014-03-14' '2014-03-15'
 '2014-03-16' '2014-03-17' '2014-03-18' '2014-03-19' '2014-03-20'
 '2014-03-21' '2014-03-22' '2014-03-23' '2014-03-24' '2014-03-25'
 '2014-03-26' '2014-03-27' '2014-03-28' '2014-03-29' '2014-03-30'
 '2014-03-31']

Unique values in product_identifier:
 [  74  337  423  432  581  611  631  659  743  797  868  904  926
972
  973 1054 1135 1173 1190 1196 1228 1240 1242 1275 1322 1328 1365 1424
 1472 1508 1542 1548 1599 1629 1672 1694 1727 1753 2294 2332 2492 2768
 2794 2818 2853 2932 2935 3004 3008 3021]

Unique values in department_identifier:
 [11 12 21 22 31 33]

Unique values in category_of_product:
 ['others' 'fast_moving_consumer_goods' 'drinks_and_food']

Unique values in outlet:
 [111 112 113 114 221 222 223 331 332 333]

Unique values in state:
 ['Maharashtra' 'Telangana' 'Kerala']

Unique values in sales:
 [  0.   1.   3.   2.   9.   5.   8.  18.  12.  28.   4.   6.  27.
7.
  10.  47.  13.  11.  32.  23.  14.  16.  19.  33.  17.  26.  20.  87.
  35.  54.  22.  15.  36.  31.  21.  25.  34.  43.  53.  38.  30.  85.
  57. 100.  52.  24.  29.  58.  63.  37.  42.  45.  39.  72.  40.  80.
  98.  56.  70.  50.  86.  46.  51.  66.  41.  64.  81.  79.  82.  69.
  48.  94.  44.  60.  68.  49.  77.  61.  76. 119.  92.  93.  62. 124.
  59.  74.  73. 174.  96.  71. 132. 101.  55.  78.  75.  65. 105. 120.
 150. 173.  67.  83.  95. 170. 116.  90. 108. 139.  84. 293.  88. 241.
 102.  97. 156. 114. 126. 109. 118. 111.  99. 171. 242. 121. 215.  89.
  nan]

Unique values in source:
 ['train' 'test']
```

```
Unique values in id:
 [       nan 1.0000e+00 2.0000e+00 ... 1.5498e+04 1.5499e+04
1.5500e+04]

Unique values in target:
 [nan None]
```

```
print(combined_df.isnull().sum())
```

```
date                           0
product_identifier             0
department_identifier          0
category_of_product            0
outlet                         0
state                          0
sales                      15500
source                         0
id                        395000
target                    410500
dtype: int64
```

```
print("Number of duplicate rows:", combined_df.duplicated().sum())
```

```
Number of duplicate rows: 0
```

```
# Shows the total count of nulls in each column
print(combined_df.isnull().sum())
```

```
date                           0
product_identifier             0
department_identifier          0
category_of_product            0
outlet                         0
state                          0
sales                      15500
source                         0
id                        395000
target                    410500
dtype: int64
```

```
# (Optional) To see the percentage of missing values
print((combined_df.isnull().sum() / len(combined_df)) * 100)
```

```
date                         0.000000
product_identifier           0.000000
department_identifier        0.000000
category_of_product          0.000000
outlet                       0.000000
```

```
state                        0.000000
sales                        3.775883
source                       0.000000
id                          96.224117
target                     100.000000
dtype: float64
```

```
# Example: Fill numerical columns with mean
combined_df.fillna(combined_df.mean(numeric_only=True), inplace=True)

# Or drop rows with missing values (if minimal)
combined_df.dropna(inplace=True)

print("Duplicates:", combined_df.duplicated().sum())
combined_df.drop_duplicates(inplace=True)
```

```
Duplicates: 0
```

```
Empty DataFrame
Columns: [date, product_identifier, department_identifier,
category_of_product, outlet, state, sales, source, id, target]
Index: []
```

```
# Replace 'column_name' with your actual column
print(combined_df['date'].value_counts())
```

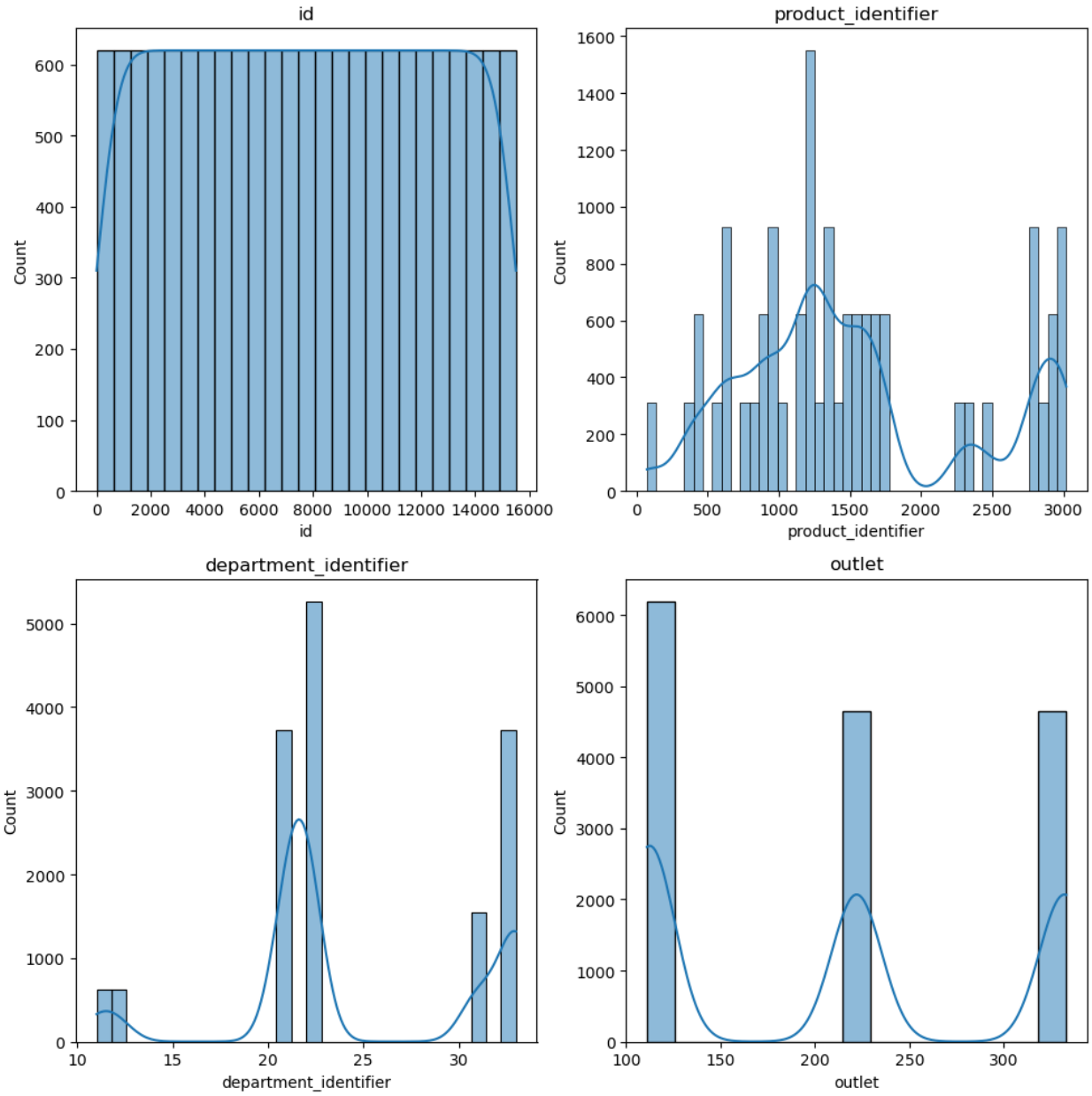
```
Series([], Name: count, dtype: int64)
```

```
# Replace 'column_name' with your actual column
print(combined_df['outlet'].value_counts())
```

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

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

# Load your data
df = pd.read_csv(r"C:\Users\mbdev\Desktop\train_data.csv\
test_data.csv")

# Define numeric columns
num_cols = df.select_dtypes(include=['int64', 'float64']).columns

# Plotting example
plt.figure(figsize=(10,10))
plotnum = 1
for col in num_cols[:4]:  # just first 4 for demo
    plt.subplot(2, 2, plotnum)
    sns.histplot(df[col], kde=True)
```

```
    plt.title(col)
    plotnum += 1
plt.tight_layout()
plt.show()
```

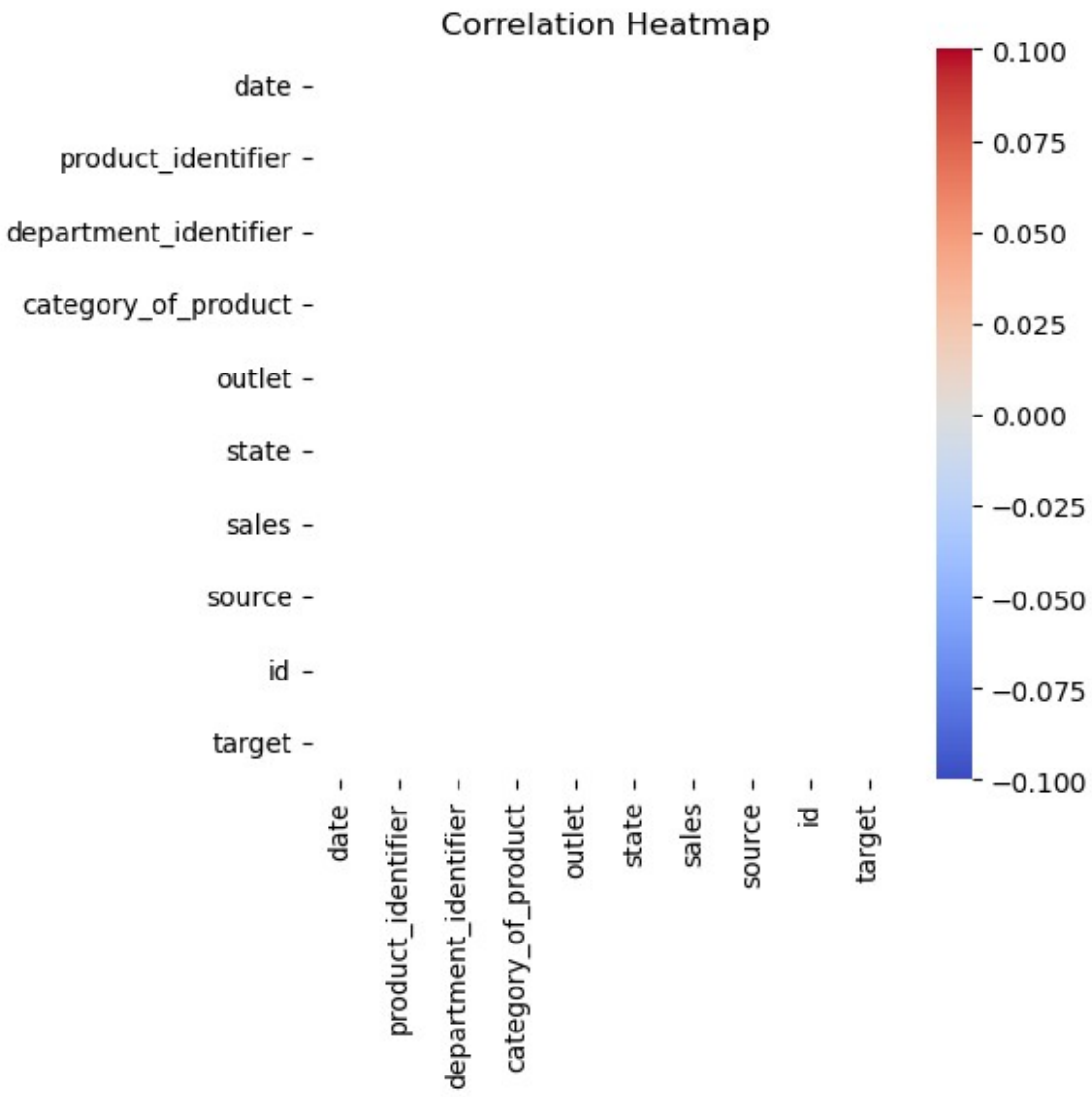



```
# Check for null values
print("Null values in each column:\n", combined_df.isnull().sum())

# Drop duplicates
print("Duplicate rows:", combined_df.duplicated().sum())
combined_df.drop_duplicates(inplace=True)
```

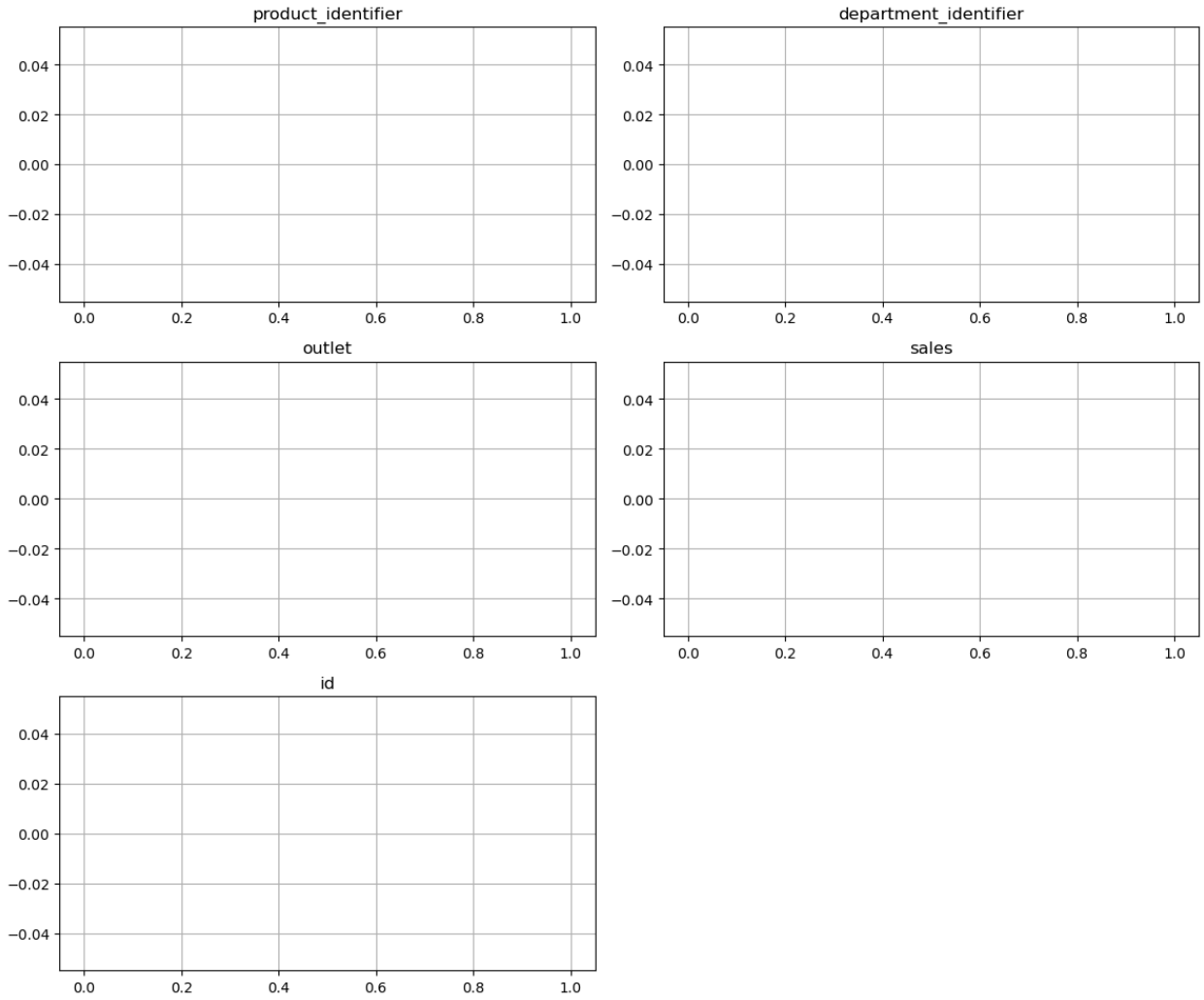
```python
# Check data types
print("\nData types:\n", combined_df.dtypes)
```

```
Null values in each column:
 date                    0
product_identifier       0
department_identifier    0
category_of_product      0
outlet                   0
state                    0
sales                    0
source                   0
id                       0
target                   0
dtype: int64
Duplicate rows: 0

Data types:
 date                     object
product_identifier        int64
department_identifier     int64
category_of_product      object
outlet                    int64
state                    object
sales                   float64
source                   object
id                      float64
target                   object
dtype: object
```

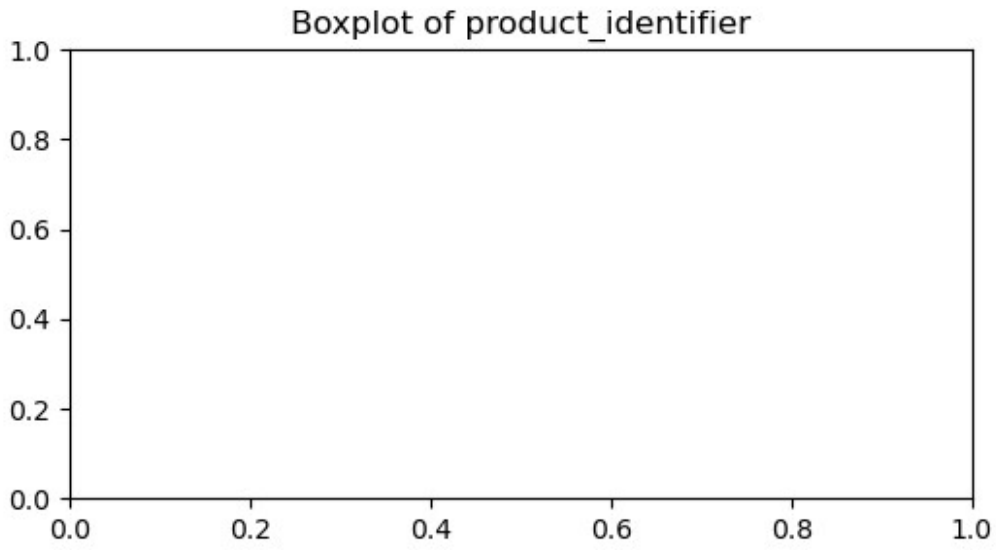
```python
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(5,5))
sns.heatmap(combined_df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```

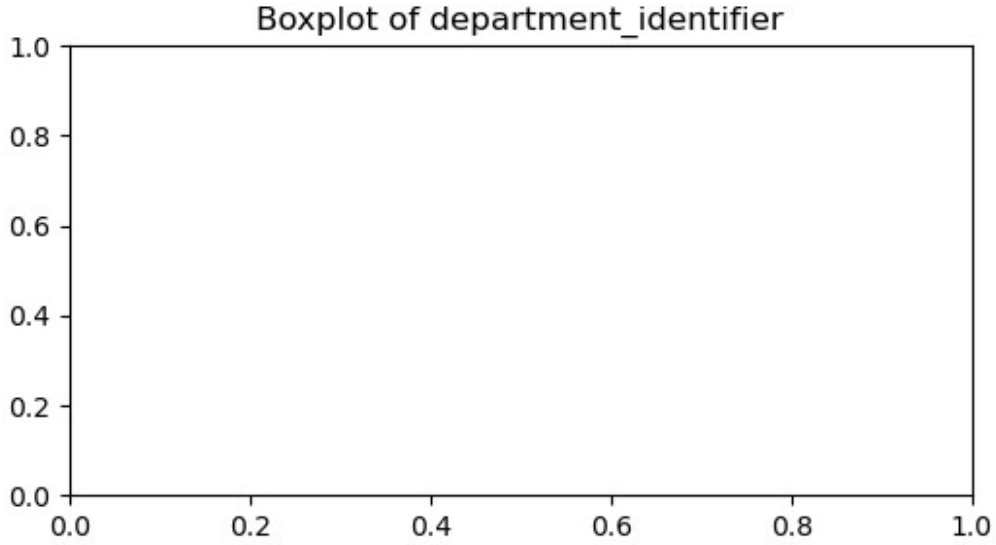


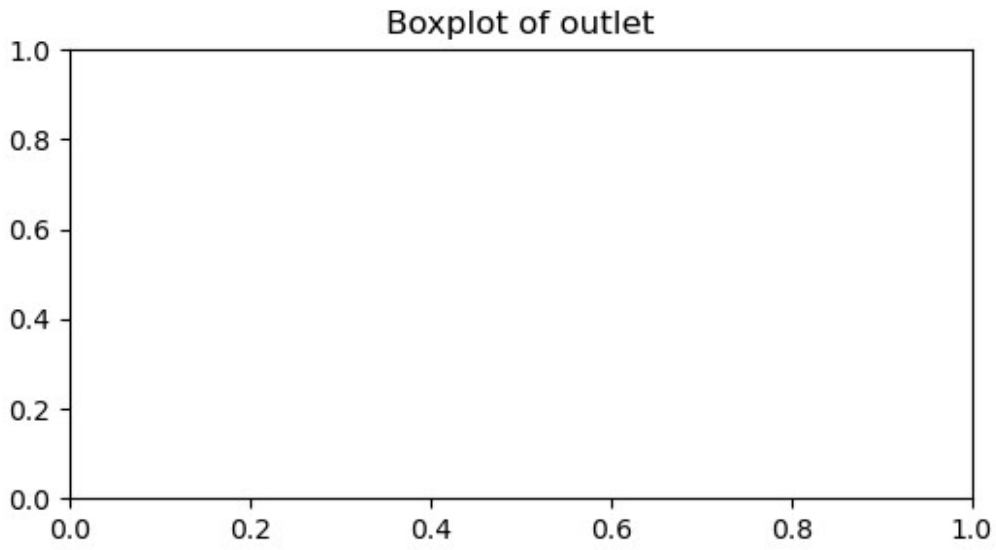

```
combined_df.hist(figsize=(12, 10))
plt.tight_layout()
plt.show()
```

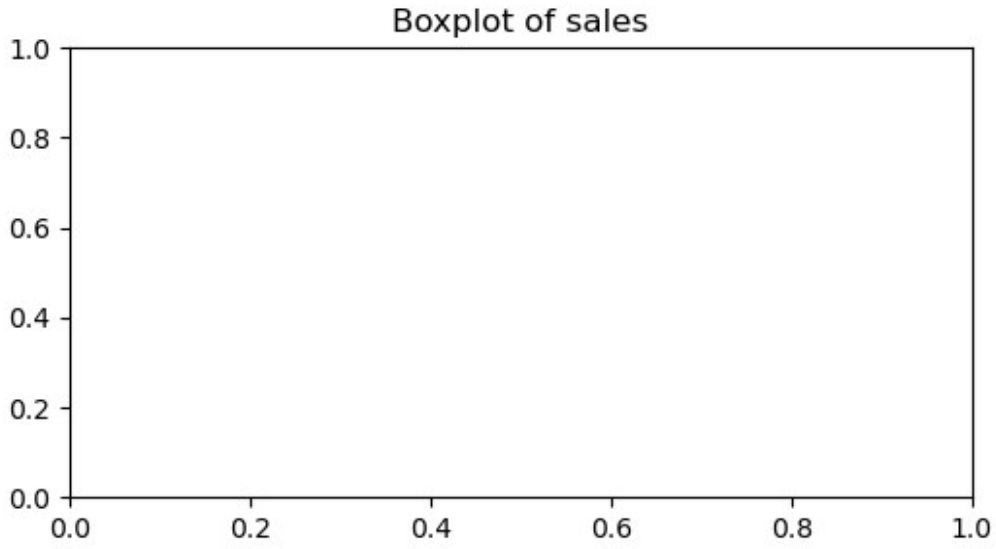


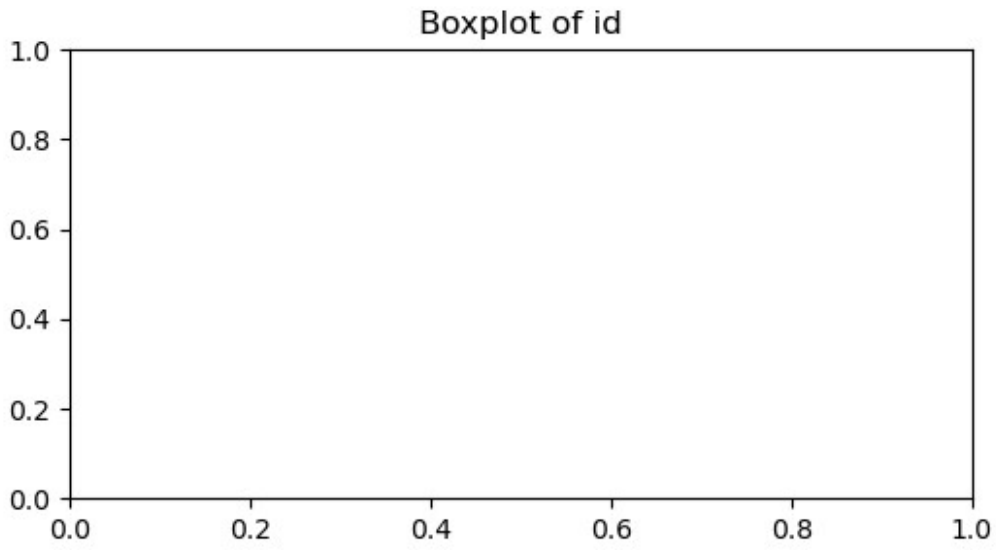

```
num_cols = combined_df.select_dtypes(include=['int64',
'float64']).columns

for col in num_cols:
    plt.figure(figsize=(6, 3))
    sns.boxplot(data=combined_df, x=col)
    plt.title(f"Boxplot of {col}")
    plt.show()
```


Boxplot of product_identifier
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Boxplot of department_identifier
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

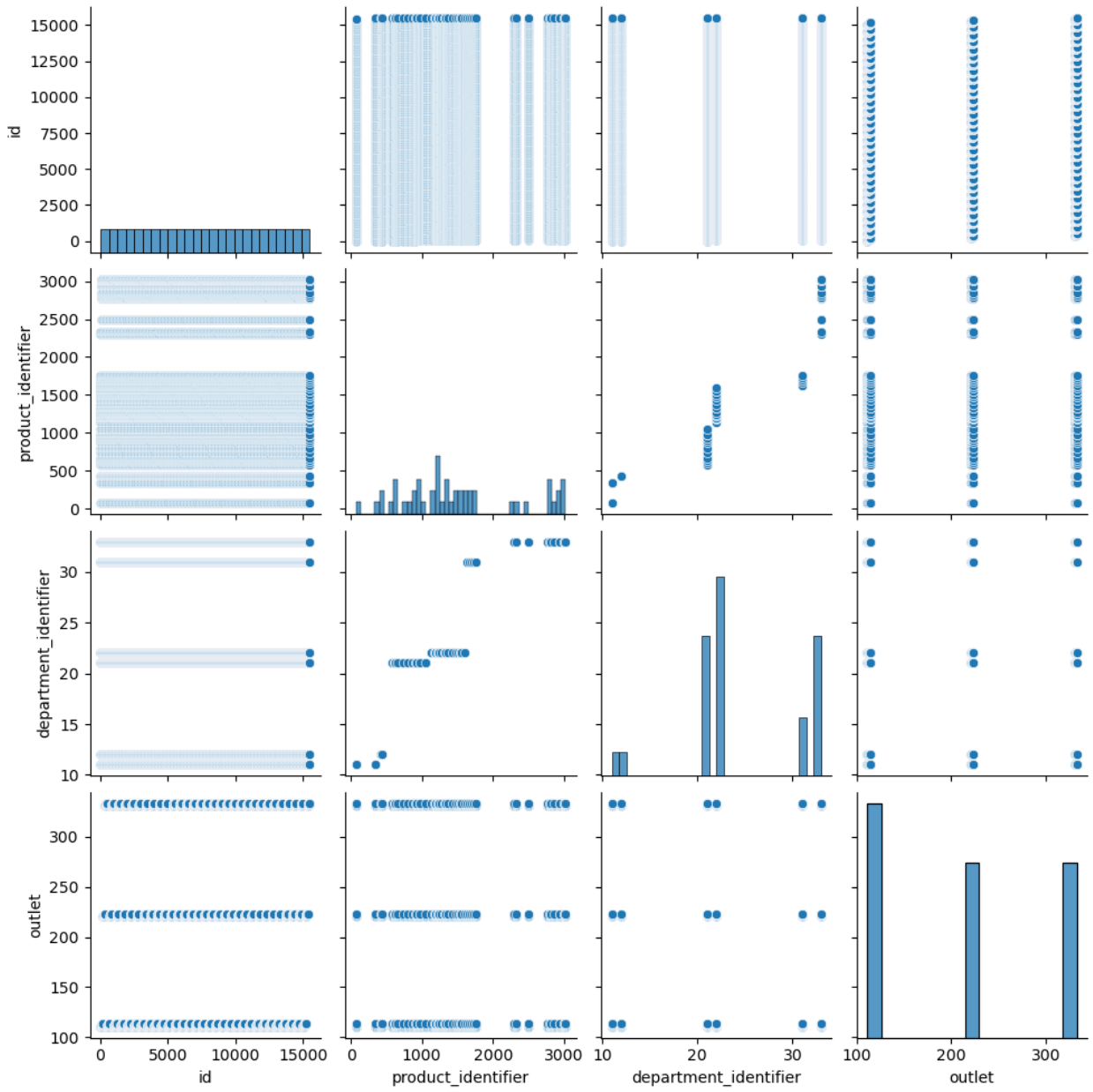

Boxplot of outlet
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Boxplot of sales
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0




```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x2b6a8cec860>
```




```
import matplotlib.pyplot as plt
import seaborn as sns

numeric_cols = combined_df.select_dtypes(include=['int64',
'float64']).columns

plt.figure(figsize=(15, 10))
for i, col in enumerate(numeric_cols[:6], 1):  # Adjust 6 based on
number of columns you want to visualize
    plt.subplot(2, 3, i)
    sns.boxplot(data=combined_df, y=col)
    plt.title(f"Boxplot of {col}")
```

```
plt.tight_layout()
plt.show()
```