In [1]:
```
from ds_helper import visualize
```

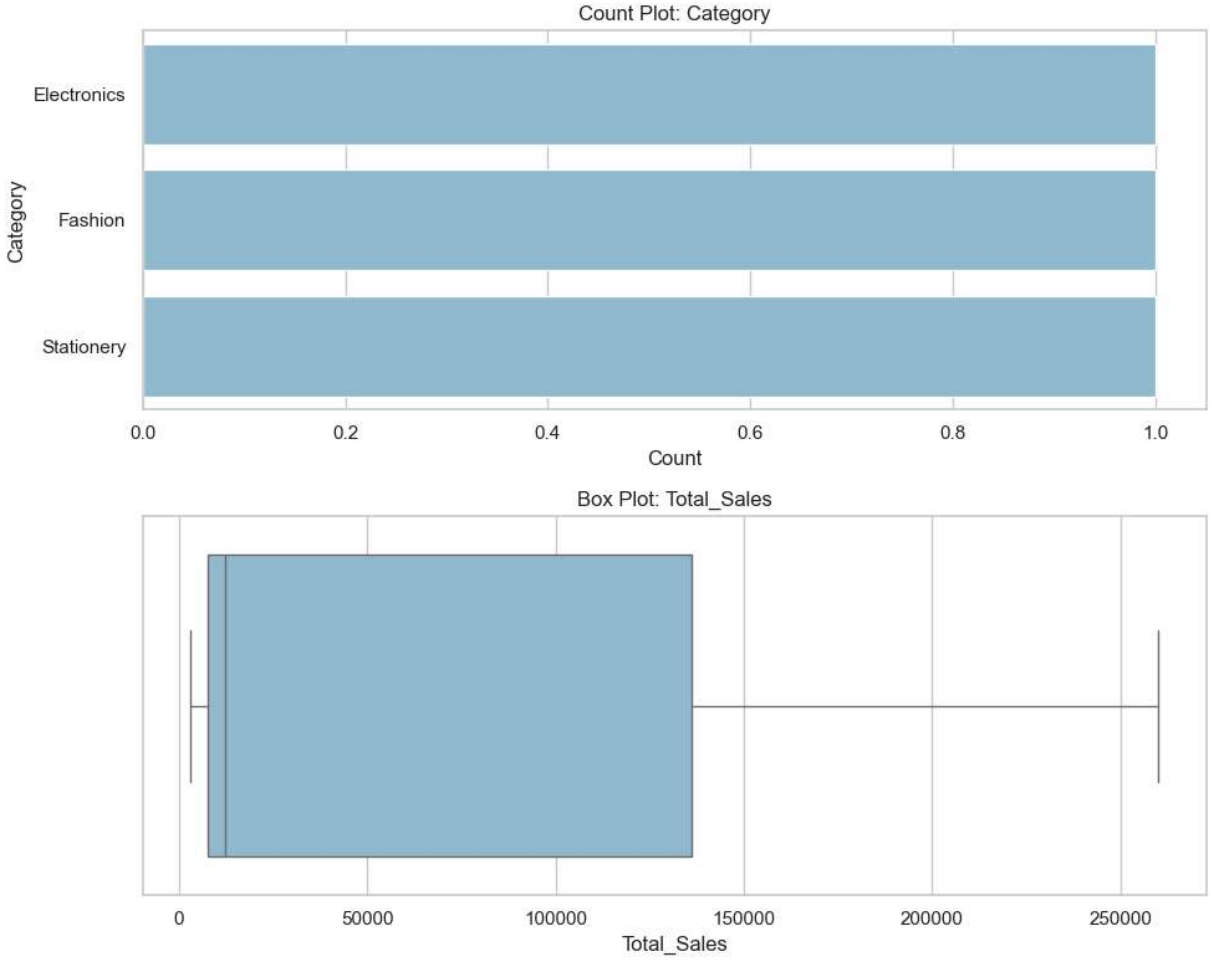
In [2]:
```
import pandas as pd
import mysql.connector

conn = mysql.connector.connect(
    host='localhost',
    user='root',
    password='GU$%ja20267',
    database='retailsales'
)
```

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
def visualize(df):
    sns.set(style="whitegrid", palette="Blues_d")
    plt.figure(figsize=(10, 8))

    # Count Plot
    plt.subplot(2, 1, 1)
    sns.countplot(y=df.columns[0], data=df)
    plt.title(f'Count Plot: {df.columns[0]}')
    plt.xlabel('Count')
    plt.ylabel(df.columns[0])

    # Box Plot
    plt.subplot(2, 1, 2)
    sns.boxplot(x=df.columns[1], data=df)
    plt.title(f'Box Plot: {df.columns[1]}')
    plt.xlabel(df.columns[1])

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

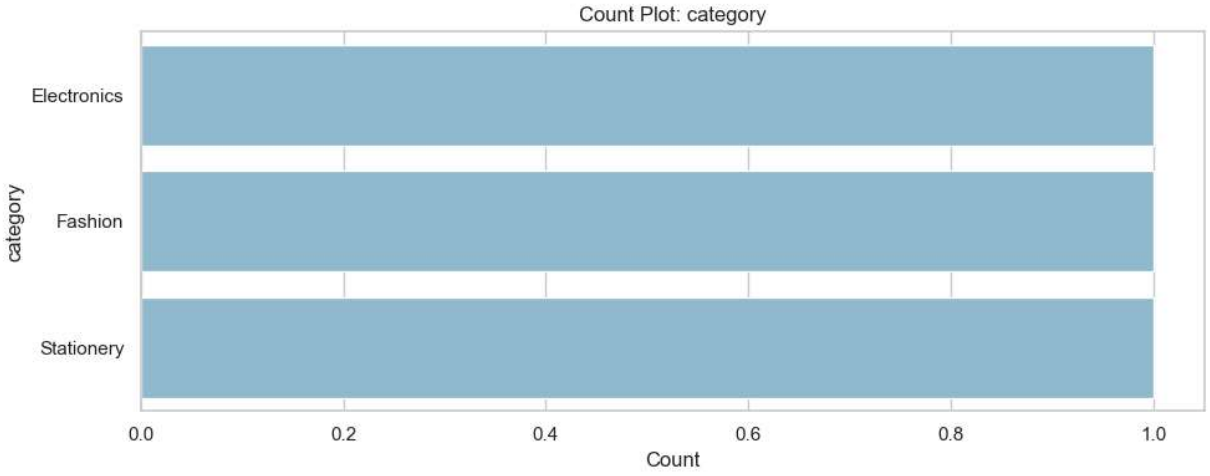
In [5]:
```
query = """
SELECT
    P.category AS Category,
    SUM(S.quantity * P.price) AS Total_Sales
FROM Sales S
JOIN Products P ON S.product_id = P.product_id
GROUP BY P.category
ORDER BY Total_Sales DESC;
"""

df = pd.read_sql(query, conn)
visualize(df)
```

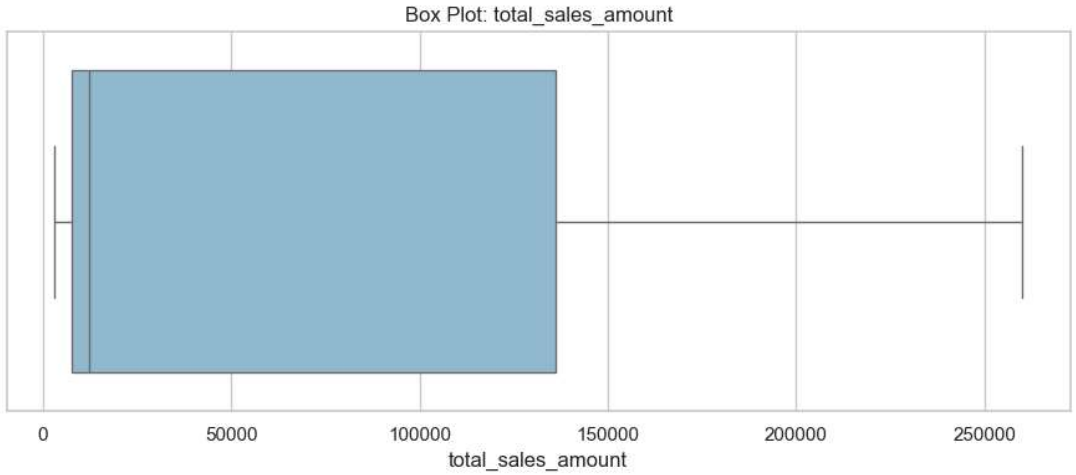
```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_2768\2191751131.py:11: UserWarning: pan
das only supports SQLAlchemy connectable (engine/connection) or database string URI
or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider u
sing SQLAlchemy.
  df = pd.read_sql(query, conn)
```



In [6]:
```
query1 = """
SELECT
    p.category,
    SUM(s.quantity * p.price) AS total_sales_amount
FROM Sales s
JOIN Products p ON s.product_id = p.product_id
GROUP BY p.category
ORDER BY total_sales_amount DESC;
"""

df1 = pd.read_sql(query1, conn)
visualize(df1)
```

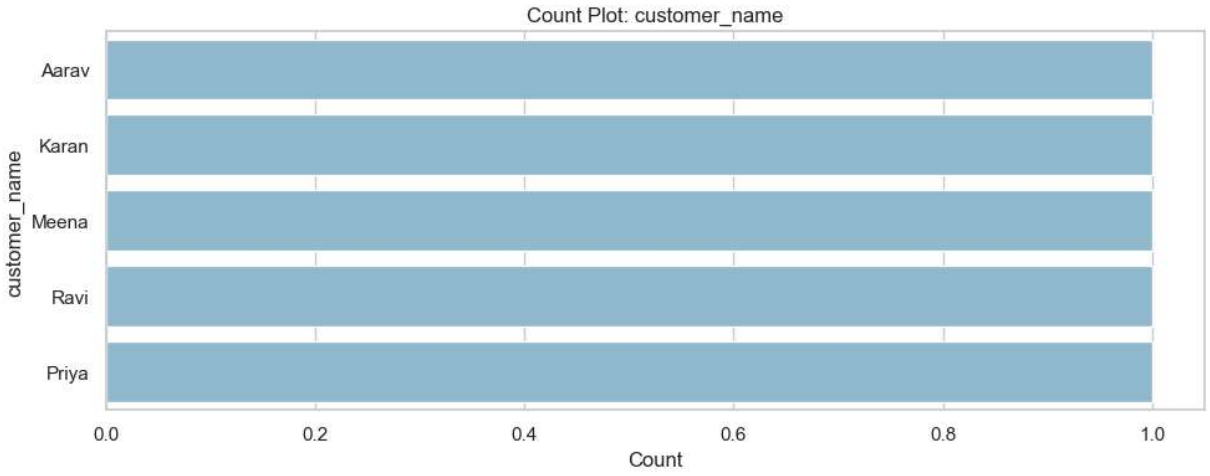
```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_2768\533635071.py:11: UserWarning: pand
as only supports SQLAlchemy connectable (engine/connection) or database string URI o
r sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider us
ing SQLAlchemy.
  df1 = pd.read_sql(query1, conn)
```

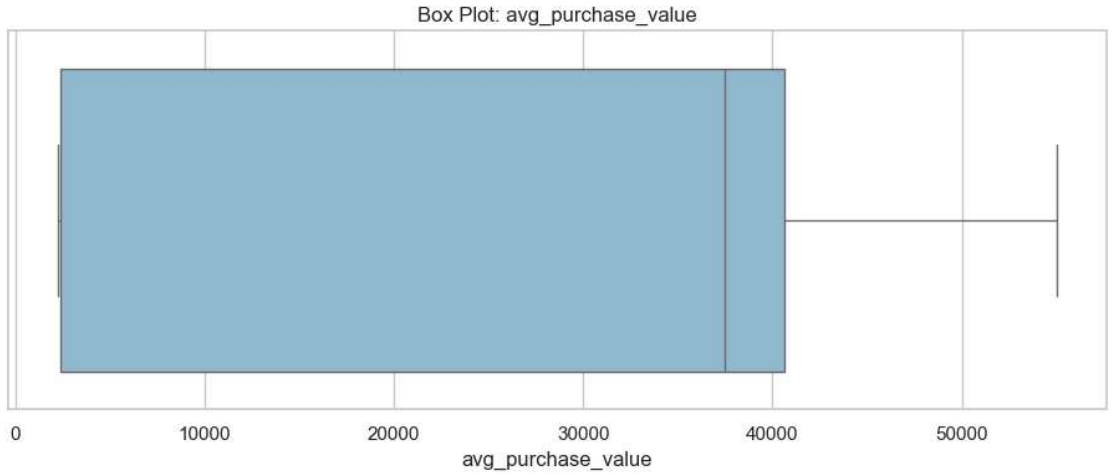




In [7]:
```
query2 = """
SELECT
    c.customer_name,
    ROUND(AVG(s.quantity * p.price), 2) AS avg_purchase_value
FROM Sales s
JOIN Products p ON s.product_id = p.product_id
JOIN Customers c ON s.customer_id = c.customer_id
GROUP BY c.customer_name
ORDER BY avg_purchase_value DESC;
"""

df2 = pd.read_sql(query2, conn)
visualize(df2)
```

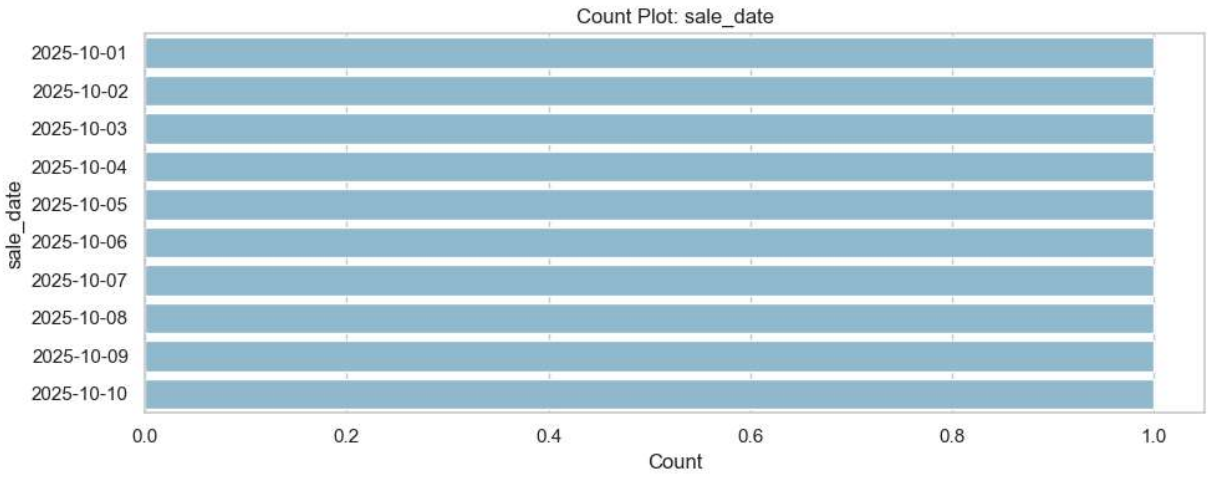
```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_2768\2669777988.py:12: UserWarning: pan
das only supports SQLAlchemy connectable (engine/connection) or database string URI
or sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider u
sing SQLAlchemy.
  df2 = pd.read_sql(query2, conn)
```

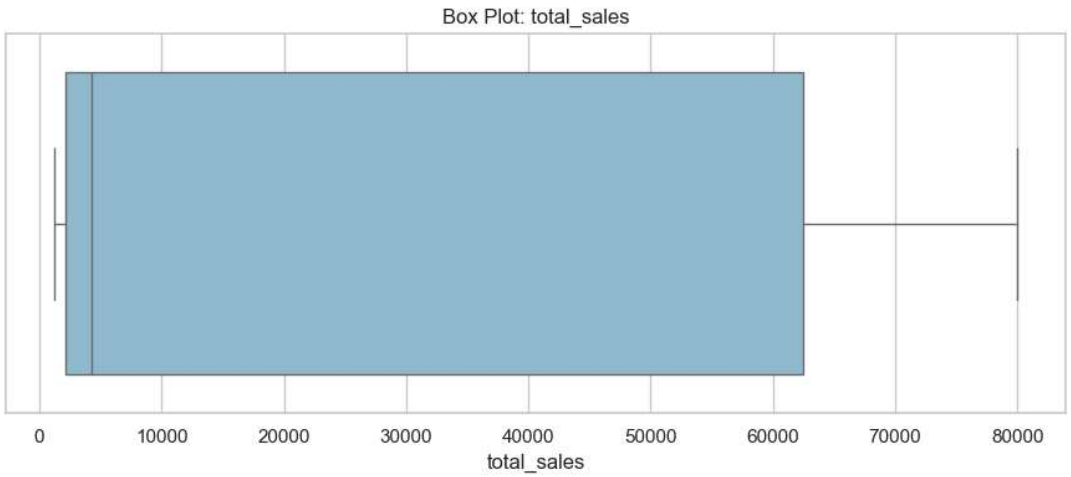




In [8]:
```
query3 = """
SELECT
    s.sale_date,
    SUM(s.quantity * p.price) AS total_sales
FROM Sales s
JOIN Products p ON s.product_id = p.product_id
GROUP BY s.sale_date
ORDER BY s.sale_date;
"""

df3 = pd.read_sql(query3, conn)
visualize(df3)
```

```
C:\Users\lenovo\AppData\Local\Temp\ipykernel_2768\93979018.py:11: UserWarning: panda
s only supports SQLAlchemy connectable (engine/connection) or database string URI or
sqlite3 DBAPI2 connection. Other DBAPI2 objects are not tested. Please consider usin
g SQLAlchemy.
  df3 = pd.read_sql(query3, conn)
```


Count Plot: sale_date
sale_date
2025-10-01
2025-10-02
2025-10-03
2025-10-04
2025-10-05
2025-10-06
2025-10-07
2025-10-08
2025-10-09
2025-10-10
0.0
0.2
0.4
0.6
0.8
1.0
Count


Box Plot: total_sales
0
10000
20000
30000
40000
50000
60000
70000
80000
total_sales