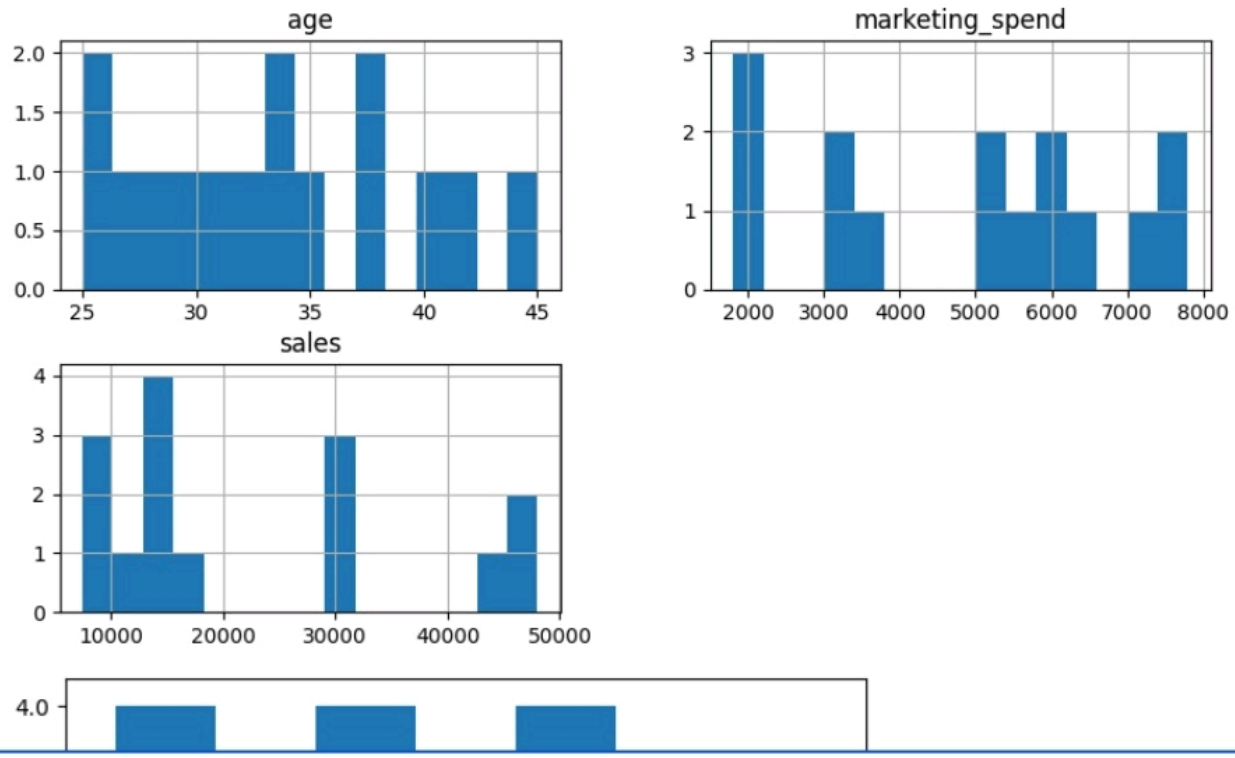
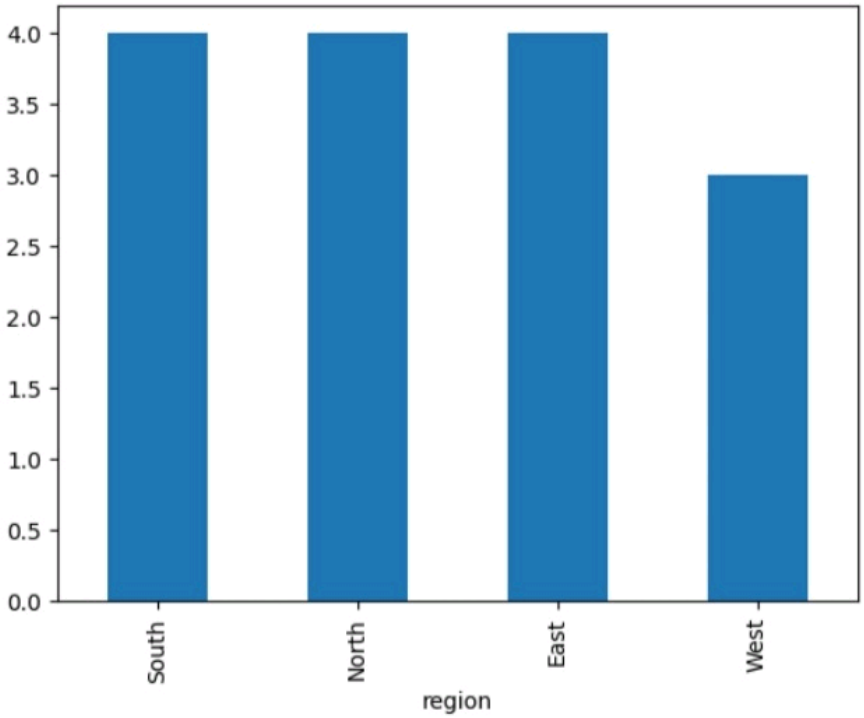
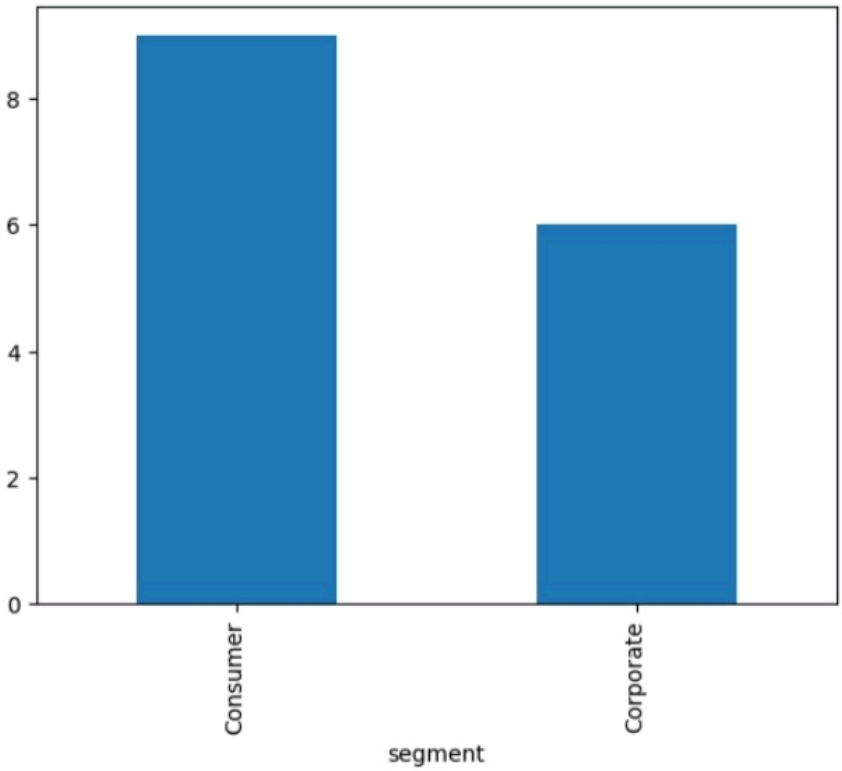
Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

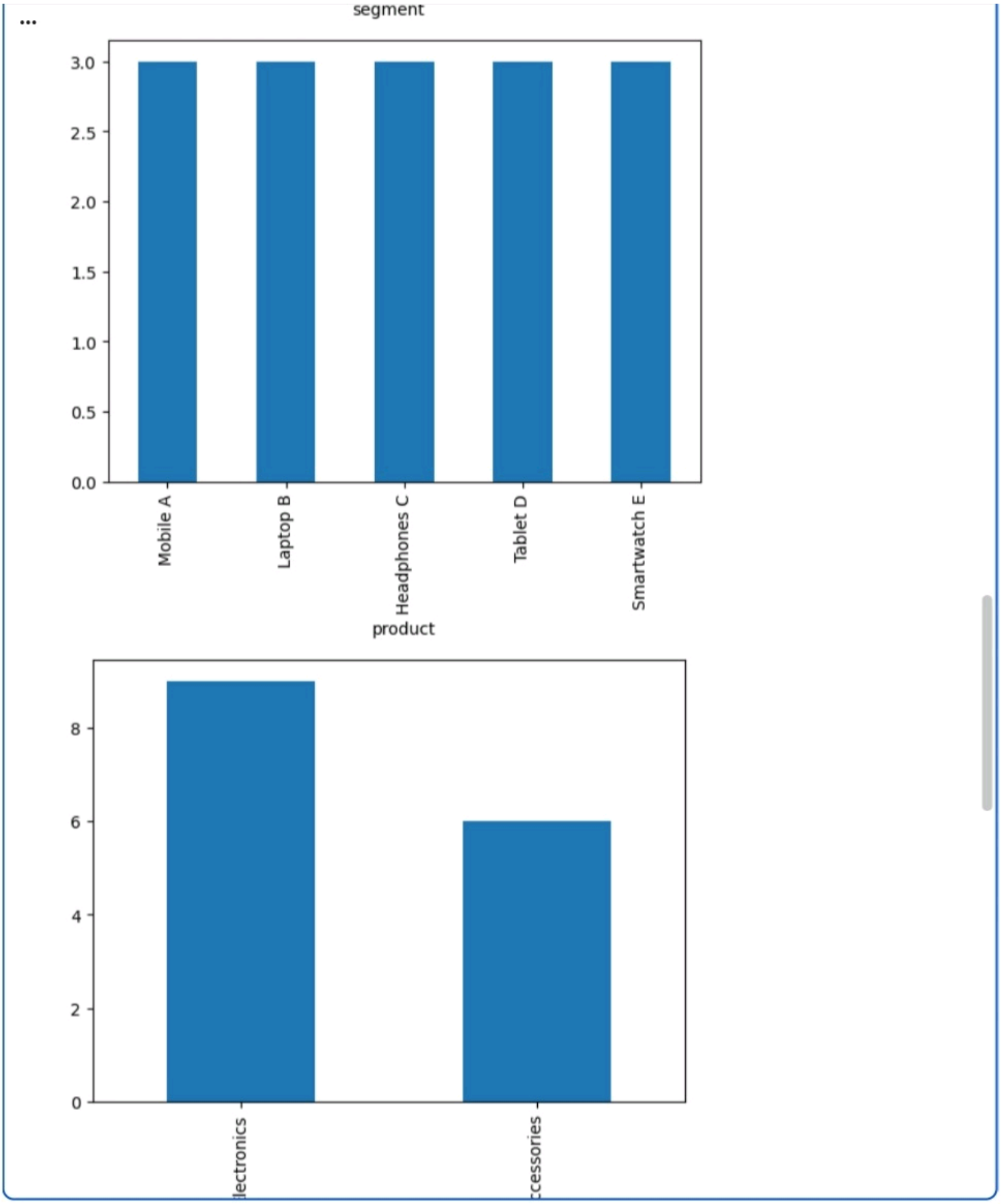
```
Saving sales_data.xlsx to sales_data (1).xlsx
   order_id  order_date customer_id  age region    segment       product  \
0       101  2024-01-10        C001   25  South   Consumer      Mobile A
1       102  2024-01-15        C002   32  North  Corporate      Laptop B
2       103  2024-02-05        C003   28   East   Consumer  Headphones C
3       104  2024-02-20        C004   40   West  Corporate      Tablet D
4       105  2024-03-02        C005   35  South   Consumer  Smartwatch E

      category  marketing_spend  sales
0  Electronics             5000  15000
1  Electronics             7000  45000
2  Accessories             2000   8000
3  Electronics             6000  30000
4  Accessories             3000  12000
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15 entries, 0 to 14
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   order_id         15 non-null     int64
 1   order_date       15 non-null     object
 2   customer_id      15 non-null     object
 3   age              15 non-null     int64
 4   region           15 non-null     object
 5   segment          15 non-null     object
 6   product          15 non-null     object
 7   category         15 non-null     object
 8   marketing_spend  15 non-null     int64
 9   sales            15 non-null     int64
dtypes: int64(4), object(6)
memory usage: 1.3+ KB
         order_id       age  marketing_spend         sales
count   15.000000  15.00000        15.000000     15.000000
mean   108.000000  33.40000      4780.000000  22746.666667
std      4.472136   5.94979      2056.418523  14636.249845
min    101.000000  25.00000      1800.000000   7500.000000
25%    104.500000  28.50000      3150.000000  12750.000000
50%    108.000000  33.00000      5200.000000  15500.000000
75%    111.500000  37.50000      6150.000000  30750.000000
max    115.000000  45.00000      7800.000000  48000.000000
            age  marketing_spend         sales
count  15.00000        15.000000     15.000000
mean   33.40000      4780.000000  22746.666667
std     5.94979      2056.418523  14636.249845
min    25.00000      1800.000000   7500.000000
25%    28.50000      3150.000000  12750.000000
50%    33.00000      5200.000000  15500.000000
75%    37.50000      6150.000000  30750.000000
max    45.00000      7800.000000  48000.000000
region
region
South    4
North    4
East     4
West     3
```

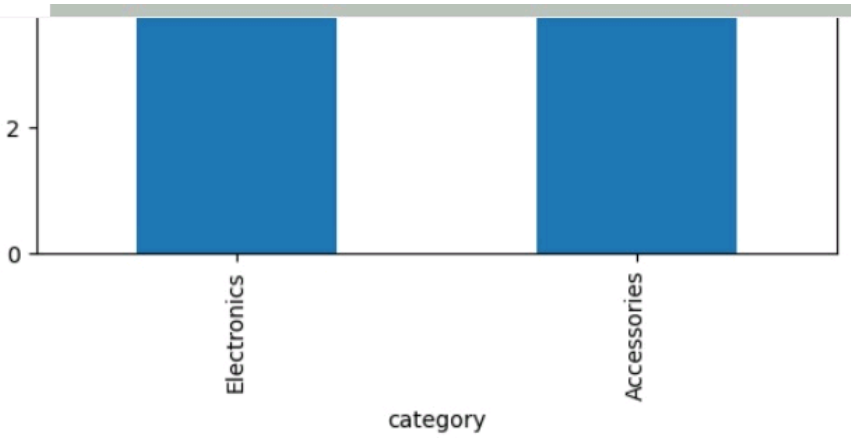
```
min    25.00000      1800.000000   7500.000000
25%    28.50000      3150.000000  12750.000000
50%    33.00000      5200.000000  15500.000000
75%    37.50000      6150.000000  30750.000000
max    45.00000      7800.000000  48000.000000
region
region
South    4
North    4
East     4
West     3
Name: count, dtype: int64
segment
segment
Consumer     9
Corporate    6
Name: count, dtype: int64
product
product
Mobile A        3
Laptop B        3
Headphones C    3
Tablet D        3
Smartwatch E    3
Name: count, dtype: int64
category
category
Electronics    9
Accessories    6
Name: count, dtype: int64
```

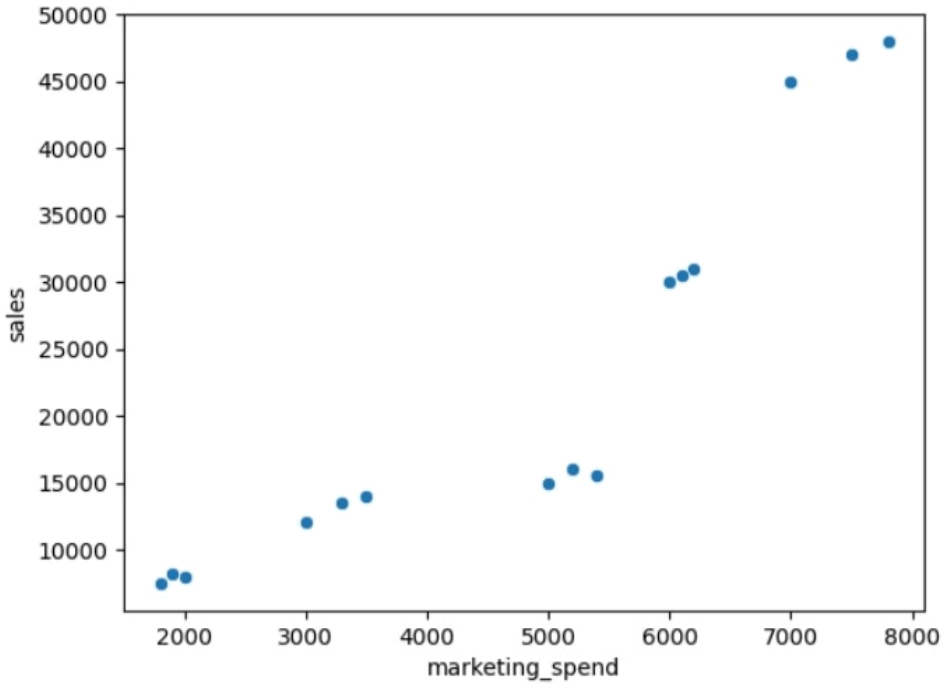


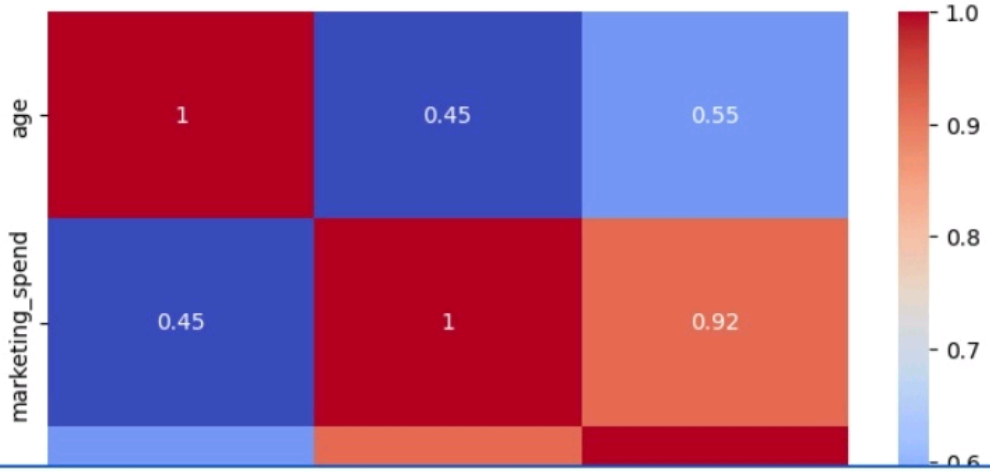

4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
South
North
East
West
region


8
6
4
2
0
Consumer
Corporate
segment


segment
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Mobile A
Laptop B
Headphones C
Tablet D
Smartwatch E
product
8
6
4
2
0
lectronics
ccessories


2
0
Electronics
Accessories
category


50000
45000
40000
35000
30000
25000
20000
15000
10000
sales
2000
3000
4000
5000
6000
7000
8000
marketing_spend


age
marketing_spend
1
0.45
0.55
0.45
1
0.92
1.0
0.9
0.8
0.7
0.6

usiness Intelligence

Edit View Insert Runtime Tools Help

Share

ds + Code + Text Run all Connect

```python
import sqlite3
import pandas as pd
```

```python
# Connect to a database (creates if not exists)
conn = sqlite3.connect("sales_data.db")
cur = conn.cursor()
```

```python
cur.execute("""
CREATE TABLE IF NOT EXISTS sales_data (
  order_id INTEGER,
  order_date TEXT,
  customer_id TEXT,
  age INTEGER,
  region TEXT,
  segment TEXT,
  product TEXT,
  category TEXT,
  marketing_spend INTEGER,
  sales INTEGER
)
""")
```

```
<sqlite3.Cursor at 0x796e96140640>
```

```python
data = [
    (101,"2024-01-10","C001",25,"South","Consumer","Mobile A","Electronics",5000,15000
    (102,"2024-01-15","C002",32,"North","Corporate","Laptop B","Electronics",7000,4500
    (103,"2024-02-05","C003",28,"East","Consumer","Headphones C","Accessories",2000,80
    (104,"2024-02-20","C004",40,"West","Corporate","Tablet D","Electronics",6000,30000
    (105,"2024-03-02","C005",35,"South","Consumer","Smartwatch E","Accessories",3000,1
    (106,"2024-03-18","C006",29,"North","Consumer","Mobile A","Electronics",5200,16000
    (107,"2024-04-10","C007",45,"East","Corporate","Laptop B","Electronics",7500,47000
    (108,"2024-04-22","C008",31,"West","Consumer","Headphones C","Accessories",1800,75
    (109,"2024-05-05","C009",27,"South","Consumer","Tablet D","Electronics",6200,31000
    (110,"2024-05-19","C010",38,"North","Corporate","Smartwatch E","Accessories",3500,
    (111,"2024-06-03","C011",33,"East","Consumer","Mobile A","Electronics",5400,15500)
    (112,"2024-06-15","C012",41,"West","Corporate","Laptop B","Electronics",7800,48000
    (113,"2024-07-08","C013",26,"South","Consumer","Headphones C","Accessories",1900,8
    (114,"2024-07-21","C014",34,"North","Consumer","Tablet D","Electronics",6100,30500
    (115,"2024-08-11","C015",37,"East","Corporate","Smartwatch E","Accessories",3300,1

]
cur.executemany("INSERT INTO sales_data VALUES (?,?,?,?,?,?,?,?,?,?)", data)
conn.commit()
```

```python
# Example 1: Select all
df = pd.read_sql_query("SELECT * FROM sales_data", conn)
df.head()
```

	order_id	order_date	customer_id	age	region	segment	product	category	marketing_spe
0	101	2024-01-10	C001	25	South	Consumer	Mobile A	Electronics	5(
1	102	2024-01-15	C002	32	North	Corporate	Laptop B	Electronics	7(
2	103	2024-02-05	C003	28	East	Consumer	Headphones C	Accessories	2(
3	104	2024-02-20	C004	40	West	Corporate	Tablet D	Electronics	6(
4	105	2024-03-02	C005	35	South	Consumer	Smartwatch E	Accessories	3(

```python
# Example 2: Top 5 products by revenue
df_top = pd.read_sql_query("""
SELECT product, SUM(sales) AS revenue
FROM sales_data
GROUP BY product
ORDER BY revenue DESC
LIMIT 8
""", conn)
df_top
```

	product	revenue
0	Laptop B	140000
1	Tablet D	91500

usiness Intelligence

Edit View Insert Runtime Tools Help

Share

ds + Code + Text Run all Connect

```python
import sqlite3
import pandas as pd
```

```python
# Connect to a database (creates if not exists)
conn = sqlite3.connect("sales_data.db")
cur = conn.cursor()
```

```python
cur.execute("""
CREATE TABLE IF NOT EXISTS sales_data (
  order_id INTEGER,
  order_date TEXT,
  customer_id TEXT,
  age INTEGER,
  region TEXT,
  segment TEXT,
  product TEXT,
  category TEXT,
  marketing_spend INTEGER,
  sales INTEGER
)
""")
```

```
<sqlite3.Cursor at 0x796e96140640>
```

```python
data = [
    (101,"2024-01-10","C001",25,"South","Consumer","Mobile A","Electronics",5000,15000
    (102,"2024-01-15","C002",32,"North","Corporate","Laptop B","Electronics",7000,4500
    (103,"2024-02-05","C003",28,"East","Consumer","Headphones C","Accessories",2000,80
    (104,"2024-02-20","C004",40,"West","Corporate","Tablet D","Electronics",6000,30000
    (105,"2024-03-02","C005",35,"South","Consumer","Smartwatch E","Accessories",3000,1
    (106,"2024-03-18","C006",29,"North","Consumer","Mobile A","Electronics",5200,16000
    (107,"2024-04-10","C007",45,"East","Corporate","Laptop B","Electronics",7500,47000
    (108,"2024-04-22","C008",31,"West","Consumer","Headphones C","Accessories",1800,75
    (109,"2024-05-05","C009",27,"South","Consumer","Tablet D","Electronics",6200,31000
    (110,"2024-05-19","C010",38,"North","Corporate","Smartwatch E","Accessories",3500,
    (111,"2024-06-03","C011",33,"East","Consumer","Mobile A","Electronics",5400,15500)
    (112,"2024-06-15","C012",41,"West","Corporate","Laptop B","Electronics",7800,48000
    (113,"2024-07-08","C013",26,"South","Consumer","Headphones C","Accessories",1900,8
    (114,"2024-07-21","C014",34,"North","Consumer","Tablet D","Electronics",6100,30500
    (115,"2024-08-11","C015",37,"East","Corporate","Smartwatch E","Accessories",3300,1

]
cur.executemany("INSERT INTO sales_data VALUES (?,?,?,?,?,?,?,?,?,?)", data)
conn.commit()
```

```python
# Example 1: Select all
df = pd.read_sql_query("SELECT * FROM sales_data", conn)
df.head()
```

	order_id	order_date	customer_id	age	region	segment	product	category	marketing_spe
0	101	2024-01-10	C001	25	South	Consumer	Mobile A	Electronics	5(
1	102	2024-01-15	C002	32	North	Corporate	Laptop B	Electronics	7(
2	103	2024-02-05	C003	28	East	Consumer	Headphones C	Accessories	2(
3	104	2024-02-20	C004	40	West	Corporate	Tablet D	Electronics	6(
4	105	2024-03-02	C005	35	South	Consumer	Smartwatch E	Accessories	3(

```python
# Example 2: Top 5 products by revenue
df_top = pd.read_sql_query("""
SELECT product, SUM(sales) AS revenue
FROM sales_data
GROUP BY product
ORDER BY revenue DESC
LIMIT 8
""", conn)
df_top
```

	product	revenue
0	Laptop B	140000
1	Tablet D	91500

4	105	2024-03-02	C005	35	South	Consumer	E	Accessories	30

```
# Example 2: Top 5 products by revenue
df_top = pd.read_sql_query("""
SELECT product, SUM(sales) AS revenue
FROM sales_data
GROUP BY product
ORDER BY revenue DESC
LIMIT 8
""", conn)
df_top
```

	product	revenue
0	Laptop B	140000
1	Tablet D	91500
2	Mobile A	46500
3	Smartwatch E	39500
4	Headphones C	23700

```
# 3. Bottom 5 products by revenue
df2 = pd.read_sql_query("""
SELECT product, SUM(sales) AS revenue
FROM sales_data
GROUP BY product
ORDER BY revenue ASC
LIMIT 5
""", conn)
print("\nBottom 5 Products by Revenue")
print(df2)
```

```
Bottom 5 Products by Revenue
        product  revenue
0  Headphones C    47400
1  Smartwatch E    79000
2      Mobile A    93000
3      Tablet D   183000
4      Laptop B   280000
```

```
# 4. Average order value per customer
import pandas as pd
df4 = pd.read_sql_query("""
SELECT customer_id, AVG(sales) AS avg_order_value
FROM sales_data
GROUP BY customer_id
""", conn)
print("\nAverage Order Value per Customer")
print(df4)
```

```
Average Order Value per Customer
   customer_id  avg_order_value
0         C001          15000.0
1         C002          45000.0
2         C003           8000.0
3         C004          30000.0
4         C005          12000.0
5         C006          16000.0
6         C007          47000.0
7         C008           7500.0
8         C009          31000.0
9         C010          14000.0
10        C011          15500.0
11        C012          48000.0
12        C013           8200.0
13        C014          30500.0
14        C015          13500.0
```

```
# 5. Monthly sales trend
df5 = pd.read_sql_query("""
SELECT substr(order_date,1,7) AS month, SUM(sales) AS total_sales
FROM sales_data
GROUP BY month
ORDER BY month
""", conn)
print("\nMonthly Sales Trend")
```

4	105	2024-03-02	C005	35	South	Consumer	E	Accessories	30

```
# Example 2: Top 5 products by revenue
df_top = pd.read_sql_query("""
SELECT product, SUM(sales) AS revenue
FROM sales_data
GROUP BY product
ORDER BY revenue DESC
LIMIT 8
""", conn)
df_top
```

	product	revenue
0	Laptop B	140000
1	Tablet D	91500
2	Mobile A	46500
3	Smartwatch E	39500
4	Headphones C	23700

```
# 3. Bottom 5 products by revenue
df2 = pd.read_sql_query("""
SELECT product, SUM(sales) AS revenue
FROM sales_data
GROUP BY product
ORDER BY revenue ASC
LIMIT 5
""", conn)
print("\nBottom 5 Products by Revenue")
print(df2)
```

```
Bottom 5 Products by Revenue
        product  revenue
0  Headphones C    47400
1  Smartwatch E    79000
2      Mobile A    93000
3      Tablet D   183000
4      Laptop B   280000
```

```
# 4. Average order value per customer
import pandas as pd
df4 = pd.read_sql_query("""
SELECT customer_id, AVG(sales) AS avg_order_value
FROM sales_data
GROUP BY customer_id
""", conn)
print("\nAverage Order Value per Customer")
print(df4)
```

```
Average Order Value per Customer
   customer_id  avg_order_value
0         C001          15000.0
1         C002          45000.0
2         C003           8000.0
3         C004          30000.0
4         C005          12000.0
5         C006          16000.0
6         C007          47000.0
7         C008           7500.0
8         C009          31000.0
9         C010          14000.0
10        C011          15500.0
11        C012          48000.0
12        C013           8200.0
13        C014          30500.0
14        C015          13500.0
```

```
# 5. Monthly sales trend
df5 = pd.read_sql_query("""
SELECT substr(order_date,1,7) AS month, SUM(sales) AS total_sales
FROM sales_data
GROUP BY month
ORDER BY month
""", conn)
print("\nMonthly Sales Trend")
```

```
        product  revenue
0  Headphones C    47400
1  Smartwatch E    79000
2       Mobile A    93000
3       Tablet D   183000
4       Laptop B   280000
```

```python
# 4. Average order value per customer
import pandas as pd
df4 = pd.read_sql_query("""
SELECT customer_id, AVG(sales) AS avg_order_value
FROM sales_data
GROUP BY customer_id
""", conn)
print("\nAverage Order Value per Customer")
print(df4)
```

```
Average Order Value per Customer
   customer_id  avg_order_value
0         C001          15000.0
1         C002          45000.0
2         C003           8000.0
3         C004          30000.0
4         C005          12000.0
5         C006          16000.0
6         C007          47000.0
7         C008           7500.0
8         C009          31000.0
9         C010          14000.0
10        C011          15500.0
11        C012          48000.0
12        C013           8200.0
13        C014          30500.0
14        C015          13500.0
```

```python
# 5. Monthly sales trend
df5 = pd.read_sql_query("""
SELECT substr(order_date,1,7) AS month, SUM(sales) AS total_sales
FROM sales_data
GROUP BY month
ORDER BY month
""", conn)
print("\nMonthly Sales Trend")
print(df5)
```

```
Monthly Sales Trend
     month  total_sales
0  2024-01        60000
1  2024-02        38000
2  2024-03        28000
3  2024-04        54500
4  2024-05        45000
5  2024-06        63500
6  2024-07        38700
7  2024-08        13500
```



```python
# 6. Unique customers per segment
df6 = pd.read_sql_query("""
SELECT segment, COUNT(DISTINCT customer_id) AS unique_customers
FROM sales_data
GROUP BY segment
""", conn)
print("\nUnique Customers per Segment")
print(df6)
```

```
Unique Customers per Segment
     segment  unique_customers
0   Consumer                 9
1  Corporate                 6
```

```
0  Headphones C    47400
1  Smartwatch E    79000
2       Mobile A    93000
3       Tablet D   183000
4       Laptop B   280000
```

```python
# 4. Average order value per customer
import pandas as pd
df4 = pd.read_sql_query("""
SELECT customer_id, AVG(sales) AS avg_order_value
FROM sales_data
GROUP BY customer_id
""", conn)
print("\nAverage Order Value per Customer")
print(df4)
```

```
Average Order Value per Customer
   customer_id  avg_order_value
0         C001          15000.0
1         C002          45000.0
2         C003           8000.0
3         C004          30000.0
4         C005          12000.0
5         C006          16000.0
6         C007          47000.0
7         C008           7500.0
8         C009          31000.0
9         C010          14000.0
10        C011          15500.0
11        C012          48000.0
12        C013           8200.0
13        C014          30500.0
14        C015          13500.0
```

```python
# 5. Monthly sales trend
df5 = pd.read_sql_query("""
SELECT substr(order_date,1,7) AS month, SUM(sales) AS total_sales
FROM sales_data
GROUP BY month
ORDER BY month
""", conn)
print("\nMonthly Sales Trend")
print(df5)
```

```
Monthly Sales Trend
     month  total_sales
0  2024-01        60000
1  2024-02        38000
2  2024-03        28000
3  2024-04        54500
4  2024-05        45000
5  2024-06        63500
6  2024-07        38700
7  2024-08        13500
```

+ Code + Text

```python
# 6. Unique customers per segment
df6 = pd.read_sql_query("""
SELECT segment, COUNT(DISTINCT customer_id) AS unique_customers
FROM sales_data
GROUP BY segment
""", conn)
print("\nUnique Customers per Segment")
print(df6)
```

```
Unique Customers per Segment
     segment  unique_customers
0   Consumer                 9
1  Corporate                 6
```