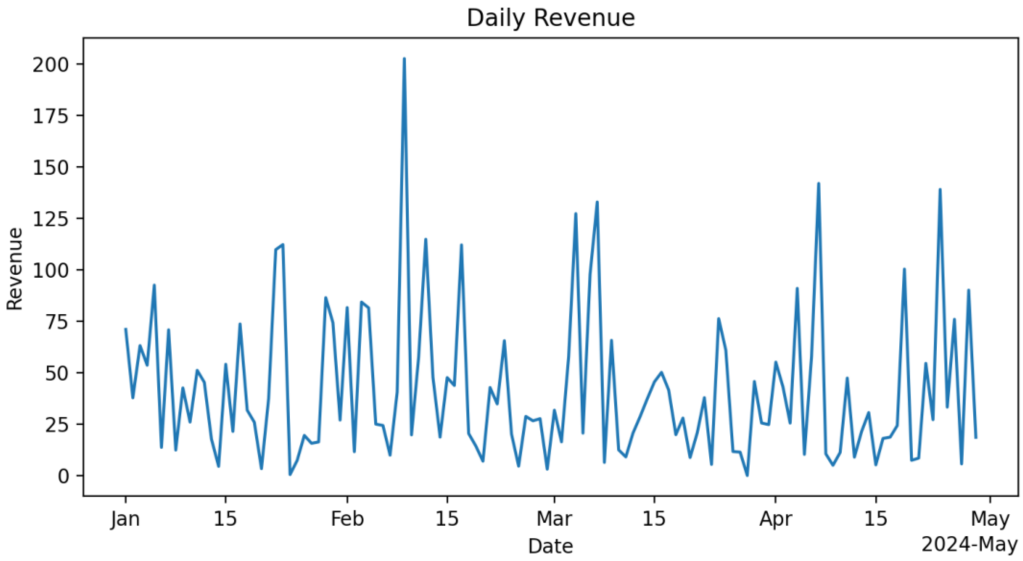
Python Code Snippet — Page 1

```python
# Personal link: https://github.com/edgarzhu7
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime, timedelta

# Ingest CSV
raw = pd.read_csv("mini_ecommerce.csv")

# Manage data types
raw["date"] = pd.to_datetime(raw["date"])
raw["user_id"] = raw["user_id"].astype("int64")
raw["channel"] = raw["channel"].astype("category")
raw["sessions"] = raw["sessions"].astype("int64")
raw["conversion_rate"] = raw["conversion_rate"].astype("float64")
raw["avg_order_value"] = raw["avg_order_value"].astype("float64")

# Wrangle
raw["orders"] = (raw["sessions"] * raw["conversion_rate"]).round(2)
raw["revenue"] = (raw["orders"] * raw["avg_order_value"]).round(2)

daily = raw.groupby("date", as_index=False).agg(
    sessions=("sessions","sum"),
    orders=("orders","sum"),
    revenue=("revenue","sum"),
)

daily["revenue_lag"] = daily["revenue"].shift(1)
daily["rev_growth"] = (daily["revenue"]/daily["revenue_lag"] - 1.0)

# Custom function
def summarize_channel_performance(frame: pd.DataFrame, top_k: int = 2) -> pd.DataFrame:
    grp = frame.groupby("channel", as_index=False).agg(
        sessions=("sessions","sum"),
        orders=("orders","sum"),
        revenue=("revenue","sum"),
        aov=("avg_order_value","mean"),
        cr=("conversion_rate","mean"),
    )
    overall_rev = grp["revenue"].sum()
    overall_cr = frame["conversion_rate"].mean()
    grp["revenue_share"] = grp["revenue"] / overall_rev
    grp["cr_lift_vs_overall"] = grp["cr"] / overall_cr - 1.0
    return grp.sort_values("revenue", ascending=False).head(top_k).round(4)
```

Python Code + Outputs — Page 2

```
# Use function + print outputs
channel_summary = summarize_channel_performance(raw, top_k=3)

print("==== Head Preview ====")
print(raw.head(5))

print("\n==== Daily Aggregation Preview ====")
print(daily.head(7)[["date","sessions","orders","revenue","rev_growth"]])

print("\n==== Channel Summary (Top 3) ====")
print(channel_summary)

# Visualization
plt.plot(daily["date"], daily["revenue"])
plt.title("Daily Revenue")
plt.show()

# --- Program Output ---
==== Head Preview ====
      date  user_id channel  sessions  conversion_rate  avg_order_value  orders  revenue
2024-01-01     1038 organic         9            0.151            52.21    1.36    71.01
2024-01-02     1028 organic         3            0.288            43.89    0.86    37.75
2024-01-03     1014 organic         5            0.301            42.09    1.50    63.14
2024-01-04     1042     ads         8            0.131            51.02    1.05    53.57
2024-01-05     1007     ads         7            0.312            42.44    2.18    92.52

==== Daily Aggregation Preview ====
      date  sessions  orders  revenue  rev_growth
2024-01-01         9    1.36    71.01         NaN
2024-01-02         3    0.86    37.75   -0.468385
2024-01-03         5    1.50    63.14    0.672583
2024-01-04         8    1.05    53.57   -0.151568
2024-01-05         7    2.18    92.52    0.727086
2024-01-06         3    0.26    13.67   -0.852248
```


Daily Revenue
Revenue
Jan
Feb
Mar
Apr
May
2024-May
Date