Session 2.4: Midterm Analytics Pipeline Defense Part 2: Exploratory Findings and Executive Review

THE BUSINESS SITUATION

- An online marketplace earns a fee on every sale its sellers make, in three regions: North America, Europe, and Asia Pacific
- Big sales are not the same as cash in the bank: the region that sells the most may not be the one that generates the most cash
- The CFO (Chief Financial Officer) asks: which region turns its sales into the most cash, and where should we invest next year? Today, you answer with one script that anyone can rerun, and present your midterm findings in the same way

Today's flow: 60 min lecture, then 120 min hands-on lab

EXPECTED OUTCOMES

By the end of today, you will be able to:

1. ***"Can someone else get the same numbers?"***
   Build a reproducible pipeline (one script that goes from the raw file to the final report with no manual steps), organized into functions
2. ***"Which region turns sales into the most cash?"***
   Calculate revenue and the operating cash flow margin by region and quarter
3. ***"How do I explain it to the CFO?"***
   Write findings as finding, evidence, and recommendation; review a classmate's script; and fix a `TypeError` (mixing text and numbers)

Session Map: Follow the Numbers

Today's question: *Which region turns its sales into the most cash, and where should we invest next year?*

1 Why it matters
Lecture, part 1

1. **Session Purpose**
   answer today's question
2. **Learning Outcomes**
   3 skills you will show today
3. **Business Context**
   How a Marketplace Makes Money and Cash
4. **Data Basics**
   What Makes a Pipeline Reproducible

→

2 Learn the tools
Lecture, part 2

5. **Lab Dataset**
   1 CSV
6. **Keywords**
   6 terms
7. **Core Foundations**
   5 ideas, one at a time
8. **IDE Setup**
   name files, run in the Terminal
9. **Code Examples**
   3 demos + Quick Checks
10. **References**
    4 official docs

→

3 Practice
Lab

11. **Exercise 1: Guided**
    Quarterly Review Pipeline
12. **Exercise 2: Applied**
    Peer Review a Classmate's Script
13. **Exercise 3: Advanced**
    Where Should We Invest Next Year?

How a Marketplace Makes Money and Cash

KEY POINTS

- Four terms, one example (North America, 2024-Q1):
- OCF margin = OCF / revenue x 100.
- **Midterm Project, Step 2 of 2: present your findings.** In a short presentation, walk through:
  - Your Question Card from Session 2.3 (worry, question, hypothesis, data, baseline)
  - Your audit result: what you checked and what you fixed
  - Your findings, each written in three parts:
  - A live run of your script, from the raw file to the final report

SEE IT ON REAL DATA

Term	Meaning	Example
GMV (Gross Merchandise Value)	the total value of everything sold on the platform	`145.2` USD M
Take rate	the share of GMV the platform keeps as its fee	`12.5` %
Revenue	GMV x take rate / 100: the platform's own income	`18.15` USD M
OCF (Operating Cash Flow)	the cash the business actually generated after paying its operating costs	`5.6` USD M

What Makes a Pipeline Reproducible

KEY POINTS

- A reproducible pipeline runs from top to bottom and produces the same numbers on any computer.
- Three habits that make a script reproducible:
  - Use relative paths such as `data/...`, never paths that only exist on your computer, such as `C:/Users/Reacher/Desktop/...`
  - Calculate every number from the data; never type a result into the code by hand
  - Before presenting, delete your output files and run the script again from the start

AT A GLANCE

Stage	What it does	Example in today's script
Load	Read the raw file	`pd.read_csv("data/session_02_04_dataset.csv")`
Check	Stop if the data breaks a rule	`assert (df["Gross_Merchandise_Value_USD_M"] > 0).all()`
Calculate	Add the measures	revenue and OCF margin
Report	Print and save the result	`summary.to_csv("data/region_summary.csv")`

The Data for Today

FILES

- `data/session_02_04_dataset.csv`

Audited 12-row quarterly marketplace dataset (fictional teaching data): 3 regions x 4 quarters, one row per region and quarter:

COLUMNS

`Quarter_Code`	Reporting quarter, `2024-Q1` through `2024-Q4`.
`Market_Segment`	Region: `North_America`, `Europe`, or `Asia_Pacific`.
`Gross_Merchandise_Value_USD_M`	GMV (Gross Merchandise Value) in millions of U.S. Dollars.
`Platform_Take_Rate_Pct`	The platform's fee as a percentage of GMV.
`Operating_Cash_Flow_USD_M`	OCF (Operating Cash Flow) in millions of U.S. Dollars.

6 Terms You Need Today

TERM 1

`Reproducible Pipeline`

What: one script that goes from the raw file to the final report, with no manual steps

Why: anyone can rerun it and check your numbers

TERM 2

`Function` (`def` / `return`)

What: a named block of code that takes inputs and gives back a result

Why: each pipeline stage gets a clear name, and you can reuse it

TERM 3

`Relative Path`

What: a file path that starts from the project folder, such as `data/session_02_04_dataset.csv`

Why: works on every computer that has the project folder

TERM 4

`Data Type` and `TypeError`

What: text (`str`) and numbers (`float`, `int`) are different types; `"Total: " + 204.3` raises a `TypeError`

Why: printing a result is where this mistake happens most

TERM 5

`Executive Finding`

What: a finding, its evidence, and a recommendation

Why: managers act on clear statements backed by numbers, not on code

TERM 6

`Peer Review`

What: a classmate checks your script against a short checklist

Why: a second reader catches mistakes the author no longer sees

How It Works

7.1

Functions

Remember: `def` to define, `return` to hand back, and store the result when you call it.

7.2

A reproducible pipeline

Remember: one script, top to bottom, same numbers every time.

7.3

Text and numbers do not mix

Remember: put numbers into text with an f-string, and format them for the reader.

7.4

Hard-coded numbers

Remember: read every input from the data; type a number into code only when it is a rule, and give it a name, such as `MARGIN_LIMIT = 20`.

7.5

Writing findings for managers

Remember: finding, evidence, recommendation, and every number from the data.

7.1 Functions: On Real Data

SEE IT ON REAL DATA

Call	Result	What happened
`revenue_usd_m(145.2, 12.5)`	`18.15`	North America, Q1: `145.2 x 12.5 / 100`
`revenue_usd_m(160.8, 13.0)`	`20.9` (rounded)	North America, Q4: same function, new inputs
`revenue_usd_m(df["Gross_Merchandise_Value_USD_M"], df["Platform_Take_Rate_Pct"])`	12 values	the same function works on whole columns

Remember: `def` to define, `return` to hand back, and store the result when you call it.

From GMV to Cash, Row by Row

YourName_S24_demo_1.py — Input → Process → Output

```python
# 1. Input: load the quarterly results
import pandas as pd
df = pd.read_csv("data/session_02_04_dataset.csv")

# 2. Process: revenue, then OCF margin, for every row
df["Revenue_USD_M"] = (
    df["Gross_Merchandise_Value_USD_M"] * df["Platform_Take_Rate_Pct"] / 100
).round(2)
df["OCF_Margin_Pct"] = (
    df["Operating_Cash_Flow_USD_M"] / df["Revenue_USD_M"] * 100
).round(1)

# 3. Output: the 3 rows with the lowest margin
lowest = df.sort_values("OCF_Margin_Pct").head(3)
print(lowest[["Quarter_Code", "Market_Segment", "OCF_Margin_Pct"]])
```

HOW IT WORKS

- The first calculation multiplies each row's GMV by that row's take rate and divides by 100: revenue in USD millions.
- The second divides each row's OCF by its revenue: the OCF margin, the share of revenue that became cash.
- `df.sort_values("OCF_Margin_Pct")` orders the rows from the lowest margin to the highest, and `.head(3)` keeps the 3 lowest.

READ THE OUTPUT

	Quarter_Code	Market_Segment	OCF_Margin_Pct
8	2024-Q1	Asia_Pacific	15.0
5	2024-Q2	Asia_Pacific	16.8
2	2024-Q3	Asia_Pacific	18.6

- The 3 weakest quarters all belong to Asia Pacific: in Q1, only `15.0` cents of every revenue dollar became cash.

Quick Check 1: `.pivot_table()`

NEW TOOL

`.pivot_table()`

`df.pivot_table(index="A", columns="B", values="C")` makes a grid with one row per value of `A` and one column per value of `B`, like an Excel pivot table. Why: shows all 12 margins in one readable grid.

YOUR TASK

Fill in the Blank

Add one line to part **3. Output**:

```
print(df._____(index="Market_Segment", columns="Quarter_Code", values="OCF_Margin_Pct"))
```

Check: the `Asia_Pacific` row reads `15.0`, `16.8`, `18.6`, `22.0`: its margin improved every quarter.

The Same Work as a Pipeline of Functions

YourName_S24_demo_2.py — Input → Process → Output

```python
# 1. Input: a function that loads and checks the raw file
import pandas as pd

def load_data(path):
    df = pd.read_csv(path)
    assert (df["Gross_Merchandise_Value_USD_M"] > 0).all(), "GMV must be above 0"
    return df

# 2. Process: functions that add measures and summarize by region
def add_metrics(df):
    df["Revenue_USD_M"] = (
        df["Gross_Merchandise_Value_USD_M"] * df["Platform_Take_Rate_Pct"] / 100
    ).round(2)
    return df

def summarize(df):
    summary = df.groupby("Market_Segment").agg(
        Revenue_USD_M=("Revenue_USD_M", "sum"),
        OCF_USD_M=("Operating_Cash_Flow_USD_M", "sum"),
    )
    summary["OCF_Margin_Pct"] = (
        summary["OCF_USD_M"] / summary["Revenue_USD_M"] * 100
    ).round(1)
    return summary

# 3. Output: run the stages in order, then print and save
df = load_data("data/session_02_04_dataset.csv")
df = add_metrics(df)
summary = summarize(df)
print(summary)
summary.to_csv("data/region_summary.csv")
```

HOW IT WORKS

- `load_data(path)` reads the file and checks with `assert` that every GMV is above 0; bad data stops the pipeline here (section 7.2).
- `add_metrics(df)` adds the revenue column and returns the table.
- `summarize(df)` groups by region, sums revenue and OCF, and computes each region's margin **from its totals** (section 7.5).

READ THE OUTPUT

- Asia Pacific earns the most revenue (`70.39`) but keeps the least as cash (`18.5` %).

Quick Check 2: Docstring

NEW TOOL

Docstring

a docstring, a text in triple quotes on the first line inside a function. `help(function_name)` prints it. Why: a reviewer learns what each stage does without reading its code.

YOUR TASK

Fill in the Blank

Add one line directly under `def add_metrics(df):`, indented like the line below it: `"""Add revenue in USD M from GMV and the take rate."""`. Then add `help(add_metrics)` to part **3. Output**.

Check: the Terminal prints a help page that includes `Add revenue in USD M from GMV and the take rate.`

Write the Findings for the CFO

YourName_S24_demo_3.py — Input → Process → Output

```
# 1. Input: load the data and build the region summary
import pandas as pd
df = pd.read_csv("data/session_02_04_dataset.csv")
df["Revenue_USD_M"] = (
    df["Gross_Merchandise_Value_USD_M"] * df["Platform_Take_Rate_Pct"] / 100
).round(2)
summary = df.groupby("Market_Segment").agg(
    Revenue_USD_M=("Revenue_USD_M", "sum"),
    OCF_USD_M=("Operating_Cash_Flow_USD_M", "sum"),
)

# 2. Process: margin, then the best region by margin
summary["OCF_Margin_Pct"] = (
    summary["OCF_USD_M"] / summary["Revenue_USD_M"] * 100
).round(1)
ranked = summary.sort_values("OCF_Margin_Pct", ascending=False)
best_region = ranked.index[0]
best_margin = ranked["OCF_Margin_Pct"].iloc[0]

# 3. Output: finding, evidence, recommendation
print(f"Finding: {best_region} turns the most revenue into cash.")
print(f"Evidence: its OCF margin is {best_margin}%.")
print("Recommendation: study its cost structure before expanding elsewhere.")
```

HOW IT WORKS

- Part 1 rebuilds the region summary, as in Micro-Example 2.
- `summary.sort_values("OCF_Margin_Pct", ascending=False)` puts the best margin first.
- `ranked.index[0]` reads the first row label, here the name of the best region.

READ THE OUTPUT

- The region and the margin come from the data, not from typing: if next year's file changes, the sentence changes with it.

Quick Check 3: `.idxmax()`

NEW TOOL

`.idxmax()`

`column.idxmax()` returns the row label of the largest value, in one step. Why: finds "which region?" without sorting.

YOUR TASK

Fill in the Blank

Add one line to part **3. Output**: `print("Most revenue:", summary["Revenue_USD_M"].______())`

Check: the Terminal shows `Most revenue: Asia_Pacific`. The region with the most revenue is not the one with the best margin.

Where to Look Things Up

REFERENCE

Python Tutorial: Defining Functions

https://docs.python.org/3/tutorial/controlflow.html#defining-functions

REFERENCE

PEP 257: Docstring Conventions

https://peps.python.org/pep-0257/

REFERENCE

Pandas API Reference: DataFrame.pivot_table

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.pivot_table.html

REFERENCE

Cookiecutter Data Science: A Logical, Reasonably Standardized Project Structure

https://drivendata.github.io/cookiecutter-data-science/

Three Exercises

EXERCISE 1: GUIDED

Quarterly Review Pipeline

Goal: Build the quarterly review as a reproducible pipeline

File:
`YourName_S24_defense.py`

Check: `2024-Q4` has the highest GMV (`524.4`)

EXERCISE 2: APPLIED

Peer Review a Classmate's Script

Goal: Peer review a classmate's script with the checklist

File:
`YourName_S24_practice.py`

Check: the hard-coded `0.125` fails question 3

EXERCISE 3: ADVANCED

Where Should We Invest Next Year?

Goal: Rank regions by size, cash efficiency, and trend

File:
`YourName_S24_challenge.py`

Check: `North_America` first (margin `31.0`)

Full steps for all three exercises: sections 11-13 of the course page.