FOUNDATION LESSON

WEEK 6 · LESSON 6

A reusable analytical pipeline

Validate, compute and summarise through small reusable functions.

THE MENTAL MODEL

MENTAL MODEL

The mental model

Validate, compute and summarise through small reusable functions.

RUN THE SUPPLIED MODULE

CODE WALKTHROUGH W06-L06-S03

```
01  from pathlib import Path
02  import sys
03  sys.path.insert(0,str(Path("src").resolve()))
04  from pipeline_tools import compute_metrics, summarise
05  import pandas as pd
06  d=pd.DataFrame({"entity":["A","A"],"month":["2025-01","2025-02"
↪   ],"revenue":[100,120],"cost":[70,80],"cfo":[25,35],"capex":[5,1
↪   0]})
07  result=compute_metrics(d)
08  print(result[["profit","margin","fcf"]].to_string(index=False))
```

WHAT TO TRACK

Dependency: supplied learner/src/pipeline_tools.py

PREDICT THE RESULT

Predict the output.
Explain each operation.
Pause before continuing.

RUN THE SUPPLIED MODULE · RESULT

CODE WALKTHROUGH W06-L06-S04

```
01  from pathlib import Path
02  import sys
03  sys.path.insert(0,str(Path("src").resolve()))
04  from pipeline_tools import compute_metrics, summarise
05  import pandas as pd
06  d=pd.DataFrame({"entity":["A","A"],"month":["2025-01","2025-02"
    ],"revenue":[100,120],"cost":[70,80],"cfo":[25,35],"capex":[5,1
    0]})
07  result=compute_metrics(d)
08  print(result[["profit","margin","fcf"]].to_string(index=False))
```

WHAT TO TRACK

Dependency: supplied learner/src/pipeline_tools.py

VALIDATE THE RESULT

```
 profit   margin  fcf
   30.0 0.300000   20
   40.0 0.333333   25
```

A GROUP MARGIN FROM TOTALS

CODE WALKTHROUGH W06-L06-S05

```
from pathlib import Path
import sys
sys.path.insert(0,str(Path("src").resolve()))
from pipeline_tools import compute_metrics, summarise
import pandas as pd
d=pd.DataFrame({"entity":["A","A"],"month":["2025-01","2025-02"],"revenue":[100,120],"cost":[70,80],"cfo":[25,35],"capex":[5,10]})
base=summarise(compute_metrics(d))
scenario=summarise(compute_metrics(d,cost_factor=1.1))
print("Base:",base[["profit","margin"]].round(4).to_dict())
print("Scenario:",scenario[["profit","margin"]].round(4).to_dict())
```

WHAT TO TRACK

Same supplied module; cost multiplier only

PREDICT THE RESULT

Predict the output.
Explain each operation.
Pause before continuing.

A GROUP MARGIN FROM TOTALS · RESULT

CODE WALKTHROUGH W06-L06-S06

```
from pathlib import Path
import sys
sys.path.insert(0,str(Path("src").resolve()))
from pipeline_tools import compute_metrics, summarise
import pandas as pd
d=pd.DataFrame({"entity":["A","A"],"month":["2025-01","2025-02"],"revenue":[100,120],"cost":[70,80],"cfo":[25,35],"capex":[5,10]})
base=summarise(compute_metrics(d))
scenario=summarise(compute_metrics(d,cost_factor=1.1))
print("Base:",base[["profit","margin"]].round(4).to_dict())
print("Scenario:",scenario[["profit","margin"]].round(4).to_dict())
```

WHAT TO TRACK

Same supplied module; cost multiplier only

VALIDATE THE RESULT

```
Base: {'profit': {'A': 70.0}, 'margin': {'A': 0.3182}}
Scenario: {'profit': {'A': 55.0}, 'margin': {'A': 0.25}}
```

REJECT A NEGATIVE CAPEX OUTFLOW

CODE WALKTHROUGH W06-L06-S07

```
01  from pathlib import Path
02  import sys
03  sys.path.insert(0,str(Path("src").resolve()))
04  from pipeline_tools import compute_metrics, summarise
05  import pandas as pd
06  d=pd.DataFrame({"entity":["A","A"],"month":["2025-01","2025-02"
    ],"revenue":[100,120],"cost":[70,80],"cfo":[25,35],"capex":[5,1
    0]})
07  d.loc[0,"capex"]=-5
08  compute_metrics(d)
```

WHAT TO TRACK

Reject a negative CapEx outflow

PREDICT THE RESULT

Predict the output.
Explain each operation.
Pause before continuing.

REJECT A NEGATIVE CAPEX OUTFLOW · RESULT

CODE WALKTHROUGH W06-L06-S08

```
from pathlib import Path
import sys
sys.path.insert(0,str(Path("src").resolve()))
from pipeline_tools import compute_metrics, summarise
import pandas as pd
d=pd.DataFrame({"entity":["A","A"],"month":["2025-01","2025-02"],"revenue":[100,120],"cost":[70,80],"cfo":[25,35],"capex":[5,10]})
d.loc[0,"capex"]=-5
compute_metrics(d)
```

WHAT TO TRACK

Reject a negative CapEx outflow

VALIDATE THE RESULT

```
ValueError: Revenue, cost and CapEx must be nonnegative
```

GUIDED PRACTICE

LEARNER PRACTICE

Guided practice

Using the supplied module, compute the two-month summary at cost_factor=1.2 and print profit.

PREDICT · MODIFY · RUN

01 Restate the supplied rules.

02 Predict the result.

03 Modify and run the code.

04 Compare actual output.

05 Explain any difference.

INDEPENDENT PRACTICE

LEARNER PRACTICE

Independent practice

Using the supplied module, set the first revenue to zero. Confirm its row margin is missing while profit still records costs.

PREDICT · MODIFY · RUN

01 Restate the supplied rules.

02 Predict the result.

03 Modify and run the code.

04 Compare actual output.

05 Explain any difference.

CHECK YOUR UNDERSTANDING

MENTAL MODEL

Check your understanding

Diagnostic: Should group margin be the unweighted mean of row margins when revenue differs? No. Use total profit divided by total revenue for this question.

RECAP AND NEXT STEP

Recap and next step

Reuse one validated function chain for baseline and scenarios. This formative rehearsal supplies functions for the single required final capstone.

UNDERSTAND APPLY VALIDATE EXPLAIN