FOUNDATION LESSON

WEEK 6 · LESSON 1

Percentage changes and compounded returns

A simple return divides a change by its prior positive value; consecutive returns compound by multiplication.

THE MENTAL MODEL

MENTAL MODEL

The mental model

A simple return divides a change by its prior positive value; consecutive returns compound by multiplication.

TWO OPPOSITE PRICE MOVES

CODE WALKTHROUGH W06-L01-S03

```
01  import pandas as pd
02  p=pd.Series([100.0,110.0,100.0])
03  r=p.pct_change(fill_method=None)
04  print("Returns:",r.round(6).tolist())
05  print("Cumulative:",round((1+r.iloc[1:]).prod(skipna=False)-1,6
↪   ))
```

WHAT TO TRACK

Two opposite price moves

PREDICT THE RESULT

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

TWO OPPOSITE PRICE MOVES · RESULT

CODE WALKTHROUGH W06-L01-S04

```
01  import pandas as pd
02  p=pd.Series([100.0,110.0,100.0])
03  r=p.pct_change(fill_method=None)
04  print("Returns:",r.round(6).tolist())
05  print("Cumulative:",round((1+r.iloc[1:]).prod(skipna=False)-1,6
↪   ))
```

WHAT TO TRACK

Two opposite price moves

VALIDATE THE RESULT

```
Returns: [nan, 0.1,
-0.090909]
Cumulative: 0.0
```

A MISSING INTERMEDIATE PRICE

CODE WALKTHROUGH W06-L01-S05

```
01  import pandas as pd
02  p=pd.Series([100.0,110.0,None,121.0,125.0])
03  prior=p.shift(1)
04  r=p.pct_change(fill_method=None).where((prior>0)&(p>0))
05  chain=(1+r.iloc[1:]).cumprod(skipna=False)-1
06  print("Returns:",r.round(4).tolist())
07  print("Strict chain:",chain.round(4).tolist())
08  print("Endpoint only:",p.iloc[-1]/p.iloc[0]-1)
```

WHAT TO TRACK

A missing intermediate price

PREDICT THE RESULT

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

A MISSING INTERMEDIATE PRICE · RESULT

CODE WALKTHROUGH W06-L01-S06

```
01  import pandas as pd
02  p=pd.Series([100.0,110.0,None,121.0,125.0])
03  prior=p.shift(1)
04  r=p.pct_change(fill_method=None).where((prior>0)&(p>0))
05  chain=(1+r.iloc[1:]).cumprod(skipna=False)-1
06  print("Returns:",r.round(4).tolist())
07  print("Strict chain:",chain.round(4).tolist())
08  print("Endpoint only:",p.iloc[-1]/p.iloc[0]-1)
```

WHAT TO TRACK

A missing intermediate price

VALIDATE THE RESULT

```
Returns: [nan, 0.1, nan,
nan, 0.0331]
Strict chain: [0.1, nan,
nan, nan]
Endpoint only: 0.25
```

ZERO AND NEGATIVE PRIOR VALUES

CODE WALKTHROUGH W06-L01-S07

```
01  import pandas as pd
02  p=pd.Series([0.0,10.0,-5.0,20.0])
03  prior=p.shift(1)
04  safe=p.where(p>0)/prior.where(prior>0)-1
05  print(safe.tolist())
```

WHAT TO TRACK

Zero and negative prior values

PREDICT THE RESULT

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

ZERO AND NEGATIVE PRIOR VALUES · RESULT

CODE WALKTHROUGH W06-L01-S08

```
01  import pandas as pd
02  p=pd.Series([0.0,10.0,-5.0,20.0])
03  prior=p.shift(1)
04  safe=p.where(p>0)/prior.where(prior>0)-1
05  print(safe.tolist())
```

WHAT TO TRACK

Zero and negative prior values

VALIDATE THE RESULT

```
[nan, nan, nan, nan]
```

GUIDED PRACTICE

LEARNER PRACTICE

Guided practice

For prices [50,55,60], calculate simple returns and a strict cumulative price-only return.

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

For [100,None,120,126], print guarded returns and a strict chain that remains unknown after the gap.

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: Does a ten percent rise followed by a ten percent fall return to the start? No. The second change uses a different denominator.

RECAP AND NEXT STEP

Recap and next step

State fraction versus percent, guard prior values, disable implicit filling, and compound consecutive returns under an explicit missing-data policy.

UNDERSTAND APPLY VALIDATE EXPLAIN