Module 5: LBO (Leveraged Buyout) Modeling

Welcome to the Heart of Private Equity! 🚀 💰

What is an LBO? (The Simple Version)

Imagine you want to buy a house worth $500,000, but you only have $100,000. What do you do?

You get a **mortgage** for $400,000!

Now imagine:

- You rent out the house, collecting $40,000/year in rent
- You use that rent money to pay down the mortgage
- 5 years later, you sell the house for $700,000
- After paying off the remaining mortgage ($200,000), you walk away with $500,000
- **You invested $100,000 and got back $500,000 = 5x return!** 🎉

That's an LBO! Except instead of a house, you're buying a company. And instead of rent, it's the company's cash flow that pays down the debt.

This is **EXACTLY** what private equity firms like Blackstone, KKR, and PE Club do every day!

Why LBOs Are So Powerful 💪

The Magic of Leverage:

Let's compare two scenarios buying the SAME company for $500M:

Scenario 1: All Cash (No Debt)

- You invest: $500M
- Company grows, you sell for: $700M
- Your return: $200M profit = **1.4x return**

Scenario 2: LBO (60% Debt)

- You invest: $200M equity (40%)
- Banks lend: $300M debt (60%)
- Company's cash flow pays down debt to $100M
- You sell for: $700M
- Pay off remaining debt: -$100M
- You walk away with: $600M
- Your return: $400M profit = **3.0x return!** 🚀

Same company. Same exit price. But with leverage, you made 2x MORE money!

That's why PE firms LOVE LBOs. You make money from:

1. 📈 **Company growth** (higher EBITDA)
2. 💰 **Multiple expansion** (selling at higher multiple than you bought)
3. 🎯 **Debt paydown** (using company's cash, not yours!)

The LBO Formula (PE Secret Sauce)

Here's the simple formula every PE investor knows:

```
MOIC (Multiple on Invested Capital) =
    (Exit Value - Remaining Debt) / Equity Investment

Where:
Exit Value = Exit EBITDA × Exit Multiple

And IRR answers: "What annual return gives me this MOIC over X years?"
```

Example:

- Equity invested: $200M
- Exit value: $700M
- Remaining debt: $100M
- **MOIC = ($700M - $100M) / $200M = 3.0x**
- If this happened in 5 years: **IRR ≈ 24.6%** (that's GREAT!)

PE Hurdle Rates:

- 🟢 **>25% IRR**: Home run deal! Partners get big bonuses
- 🟡 **20-25% IRR**: Good deal, do it
- 🟠 **15-20% IRR**: Okay, might pass
- 🔴 **<15% IRR**: Pass. Not worth the risk

What You'll Build in This Module

A **complete LBO model** that any PE firm would use:

The Components:

1. 📊 **Sources & Uses** - How you pay for the deal
2. 📈 **Operating Model** - Company performance over 5 years
3. 💳 **Debt Schedule** - Tracking debt paydown (this is KEY!)
4. 💵 **Cash Flow Waterfall** - Where does all the cash go?
5. 🎯 **Returns Analysis** - IRR and MOIC calculation
6. 📉 **Sensitivity Tables** - Testing different scenarios

You'll build 8 progressive files teaching each concept step-by-step!

The LBO Story: "TechCo Buyout"

Let's follow a real example throughout this module:

The Deal:

- **Target:** TechCo Inc. (SaaS company)
- **Entry EBITDA:** $50M
- **Entry Multiple:** 8.0x
- **Purchase Price:** $400M
- **Your Equity:** $160M (40%)
- **Debt:** $240M (60%)
- **Hold Period:** 5 years

The Thesis:

- EBITDA will grow from $50M → $85M (10% annual growth)
- You'll sell at 10.0x EBITDA (multiple expansion!)
- Company's cash flow will pay down debt

The Math:

- Exit value: $85M × 10.0x = $850M
- Remaining debt: ~$100M (paid down $140M!)
- Your equity at exit: $850M - $100M = $750M
- **Your return: $750M / $160M = 4.7x MOIC, 36% IRR 🚀 🚀 🚀**

Let's see if we can achieve this!

Building Your LBO: Step-by-Step

We'll build the model in **8 progressive files**:

File Structure:

1. `01_sources_and_uses.py` - How you finance the deal
2. `02_entry_valuation.py` - What you're paying (entry multiple)
3. `03_revenue_projections.py` - Company growth over 5 years
4. `04_ebitda_and_cash_flow.py` - Operating performance
5. `05_debt_schedule.py` - THE MOST IMPORTANT PART!
6. `06_exit_valuation.py` - What you sell for (exit multiple)
7. `07_returns_calculation.py` - IRR and MOIC
8. `08_sensitivity_analysis.py` - Testing different scenarios

Each file is **fun, simple, and builds on the previous one!**

Let's Build: Step 1 - Sources & Uses

Create a new file: `01_sources_and_uses.py`

```
"""
Step 1: Sources & Uses Table

This is THE FIRST thing PE firms build in an LBO.
It shows: Where is the money coming from? Where is it going?

Think of it like buying a house:
- SOURCES: Your down payment + mortgage
- USES: Purchase price + closing costs
"""

import pandas as pd

print("STEP 1: SOURCES & USES")
print("=" * 70)
print("Deal: TechCo Inc. Buyout")
print("=" * 70)

# USES (Where the money goes)
purchase_price = 400.0  # Buying the company for $400M
transaction_fees = 12.0  # Investment bankers, lawyers (3% is typical)
financing_fees = 8.0     # Bank fees for debt
total_uses = purchase_price + transaction_fees + financing_fees

uses = pd.DataFrame({
    'Item': ['Purchase Price', 'Transaction Fees', 'Financing Fees',
'TOTAL USES'],
    'Amount ($M)': [purchase_price, transaction_fees, financing_fees,
total_uses],
    '% of Total': [
        (purchase_price / total_uses) * 100,
        (transaction_fees / total_uses) * 100,
        (financing_fees / total_uses) * 100,
        100.0
    ]
})

print("\\nUSES (Where the money goes):")
print(uses.to_string(index=False))

# SOURCES (Where the money comes from)
senior_debt = 200.0      # Bank loan (cheaper, but senior in priority)
subordinated_debt = 40.0  # Mezzanine debt (more expensive, junior)
equity = 180.0           # Your money (PE fund's cash)
total_sources = senior_debt + subordinated_debt + equity

sources = pd.DataFrame({
    'Item': ['Senior Debt', 'Subordinated Debt', 'Equity', 'TOTAL
SOURCES'],
    'Amount ($M)': [senior_debt, subordinated_debt, equity,
total_sources],
    '% of Total': [
```

```
        (senior_debt / total_sources) * 100,
        (subordinated_debt / total_sources) * 100,
        (equity / total_sources) * 100,
        100.0
    ]
})

print("\\nSOURCES (Where the money comes from):")
print(sources.to_string(index=False))

# Sanity check
print("\\n" + "-" * 70)
print("BALANCE CHECK:")
print("-" * 70)
print(f"Total Sources: ${total_sources:.1f}M")
print(f"Total Uses:    ${total_uses:.1f}M")
print(f"Difference:    ${total_sources - total_uses:.1f}M")

if abs(total_sources - total_uses) < 0.01:
    print("✅ BALANCED! Sources = Uses")
else:
    print("❌ ERROR! Sources ≠ Uses")

# Key metrics
print("\\n" + "-" * 70)
print("KEY METRICS:")
print("-" * 70)
print(f"Leverage (Debt/Equity):      {(senior_debt + subordinated_debt) / equity:.2f}x")
print(f"Equity % of Purchase Price:  {(equity / purchase_price) * 100:.1f}%")
print(f"Debt % of Purchase Price:    {((senior_debt + subordinated_debt) / purchase_price) * 100:.1f}%")

print("\\n✅ Sources & Uses table complete!")
print("💡 This is how every LBO starts - know where the money comes from!")
```

Run this! You'll see how a $420M deal gets financed.

Try this:

- Change the equity % (try 30% or 50%)
- See how leverage changes
- More debt = higher returns BUT more risk!

Let's Build: Step 2 - Entry Valuation

Create a new file: `02_entry_valuation.py`

```
"""
Step 2: Entry Valuation - What Are We Paying?

In PE, you buy companies based on EBITDA multiples.
Entry Multiple × EBITDA = Purchase Price

Lower entry multiple = Better deal!
"""

import pandas as pd

print("STEP 2: ENTRY VALUATION")
print("=" * 70)

# Company financials (Last Twelve Months - LTM)
company_name = "TechCo Inc."
ltm_revenue = 250.0      # $250M revenue
ltm_ebitda = 50.0        # $50M EBITDA
ebitda_margin = (ltm_ebitda / ltm_revenue) * 100

print(f"Target: {company_name}")
print("-" * 70)
print(f"LTM Revenue:      ${ltm_revenue:.0f}M")
print(f"LTM EBITDA:       ${ltm_ebitda:.0f}M")
print(f"EBITDA Margin:    {ebitda_margin:.1f}%")

# Entry multiple
entry_multiple = 8.0
purchase_price = ltm_ebitda * entry_multiple

print("\\n" + "-" * 70)
print("VALUATION:")
print("-" * 70)
print(f"Entry Multiple:   {entry_multiple:.1f}x EBITDA")
print(f"Purchase Price:   ${purchase_price:.0f}M")
print(f"  Calculation: ${ltm_ebitda:.0f}M × {entry_multiple:.1f}x =
${purchase_price:.0f}M")

# Compare to other multiples
print("\\n" + "-" * 70)
print("IMPLIED MULTIPLES:")
print("-" * 70)

implied_ev_revenue = purchase_price / ltm_revenue
print(f"EV / Revenue:     {implied_ev_revenue:.2f}x")
print(f"EV / EBITDA:      {entry_multiple:.1f}x")

# Is this a good deal?
print("\\n" + "-" * 70)
print("IS THIS A GOOD DEAL?")
print("-" * 70)

# Typical SaaS company multiples (as of 2024-2025)
```

```
print("Typical SaaS Company Multiples:")
print("  Low growth (<10%):     6-8x EBITDA")
print("  Medium growth (10-20%): 8-12x EBITDA")
print("  High growth (>20%):    12-15x EBITDA")

if entry_multiple < 8:
    print(f"\\n✅ GREAT DEAL! Buying at {entry_multiple:.1f}x is below
market")
elif entry_multiple < 10:
    print(f"\\n🟡 FAIR DEAL. {entry_multiple:.1f}x is reasonable for this
company")
else:
    print(f"\\n🔴 EXPENSIVE! {entry_multiple:.1f}x is high - need strong
growth thesis")

# What exit multiple do we need?
print("\\n" + "-" * 70)
print("EXIT MULTIPLE TARGETS:")
print("-" * 70)

for exit_mult in [8.0, 9.0, 10.0, 11.0, 12.0]:
    multiple_expansion = exit_mult - entry_multiple
    print(f"Exit at {exit_mult:.1f}x: {multiple_expansion:+.1f}x multiple
expansion")

print("\\n✅ Entry valuation complete!")
print("💡 PE firms make money on multiple expansion - buy at 8x, sell at
10x+")
```

The PE mantra: "Buy low, sell high" - but with EBITDA multiples!

Let's Build: Step 3 - Revenue Projections

Create a new file: `03_revenue_projections.py`

```
"""
Step 3: Revenue Projections - Growing the Business

PE firms create value by growing the company.
Model realistic revenue growth over your holding period.
"""

import pandas as pd
import numpy as np

print("STEP 3: REVENUE PROJECTIONS")
print("=" * 70)

# Starting point (Year 0 = LTM)
base_revenue = 250.0  # $250M
```

```
years = list(range(2025, 2030))  # 5-year hold

# Growth strategy
print("Growth Strategy:")
print("-" * 70)
print("Year 1: 12% - Immediate improvements from PE ownership")
print("Year 2: 11% - New products launching")
print("Year 3: 10% - Market expansion")
print("Year 4: 9%  - Mature growth")
print("Year 5: 8%  - Stable growth")

growth_rates = [0.12, 0.11, 0.10, 0.09, 0.08]

# Build projections
revenues = []
current_revenue = base_revenue

for growth in growth_rates:
    current_revenue = current_revenue * (1 + growth)
    revenues.append(current_revenue)

# Create DataFrame
projections = pd.DataFrame({
    'Year': years,
    'Revenue': revenues,
    'Growth_%': [g * 100 for g in growth_rates]
})

print("\\nRevenue Projections ($M):")
print(projections.round(1))

# Calculate CAGR
cagr = ((revenues[-1] / base_revenue) ** (1/5) - 1) * 100

print("\\n" + "-" * 70)
print("SUMMARY:")
print("-" * 70)
print(f"Year 0 (LTM):     ${base_revenue:.0f}M")
print(f"Year 5 (Exit):    ${revenues[-1]:.0f}M")
print(f"Total Growth:     {((revenues[-1] / base_revenue - 1) *
100):.1f}%")
print(f"5-Year CAGR:      {cagr:.1f}%")

# Value creation
print("\\n" + "-" * 70)
print("VALUE CREATION:")
print("-" * 70)

additional_revenue = revenues[-1] - base_revenue
print(f"Additional Revenue Created: ${additional_revenue:.0f}M")
print(f"\\nIf EBITDA margin stays at 20%:")
print(f"  Additional EBITDA: ${additional_revenue * 0.20:.0f}M")
print(f"  At 10x multiple: ${additional_revenue * 0.20 * 10:.0f}M more
value! 💰 ")
```

```
print("\\n✅ Revenue projections complete!")
print("💡 PE firms grow companies through pricing, new products, M&A, and efficiency")
```

This is where PE firms add value! Growing revenue = creating value.

Let's Build: Step 4 - EBITDA & Cash Flow

Create a new file: `04_ebitda_and_cash_flow.py`

```
"""
Step 4: EBITDA and Free Cash Flow

EBITDA is the profit engine.
Free Cash Flow is what pays down debt!

FCF = EBITDA - Taxes - CapEx - Change in NWC
"""

import pandas as pd

print("STEP 4: EBITDA & FREE CASH FLOW")
print("=" * 70)

# Revenue from Step 3
years = list(range(2025, 2030))
revenues = [280, 311, 342, 373, 403]

# Operating assumptions
ebitda_margin = 0.20  # 20% EBITDA margin (typical for SaaS)
tax_rate = 0.25       # 25% corporate tax
capex_pct = 0.05      # 5% of revenue (growth capex)
nwc_pct = 0.08        # 8% of revenue

print("Operating Assumptions:")
print("-" * 70)
print(f"EBITDA Margin:    {ebitda_margin * 100:.0f}%")
print(f"Tax Rate:         {tax_rate * 100:.0f}%")
print(f"CapEx % Revenue:  {capex_pct * 100:.0f}%")
print(f"NWC % Revenue:    {nwc_pct * 100:.0f}%")

# Build operating model
operating = pd.DataFrame({
    'Year': years,
    'Revenue': revenues
})

# EBITDA
operating['EBITDA'] = operating['Revenue'] * ebitda_margin
```

```
# Taxes (simplified - on EBITDA)
operating['Taxes'] = operating['EBITDA'] * tax_rate

# CapEx
operating['CapEx'] = operating['Revenue'] * capex_pct

# Change in NWC
base_nwc = 250 * nwc_pct  # Year 0 NWC
nwc_values = operating['Revenue'] * nwc_pct
nwc_changes = nwc_values.diff()
nwc_changes.iloc[0] = nwc_values.iloc[0] - base_nwc
operating['Change_NWC'] = nwc_changes

# FREE CASH FLOW (this is what matters!)
operating['Free_Cash_Flow'] = (
    operating['EBITDA'] -
    operating['Taxes'] -
    operating['CapEx'] -
    operating['Change_NWC']
)

print("\\nOperating Model & Free Cash Flow ($M):")
print(operating.round(1))

# Summary
print("\\n" + "=" * 70)
print("FREE CASH FLOW SUMMARY:")
print("=" * 70)

total_fcf = operating['Free_Cash_Flow'].sum()
print(f"\\nTotal 5-Year FCF: ${total_fcf:.1f}M")
print(f"Average Annual FCF: ${total_fcf / 5:.1f}M")

# This cash flow goes to...
print("\\n" + "-" * 70)
print("WHERE DOES THIS CASH GO?")
print("-" * 70)
print("1. Pay interest on debt")
print("2. Pay down debt principal")
print("3. Build cash balance (if any left over)")
print("\\n💡 The more FCF, the faster you pay down debt!")
print("💡 Less debt at exit = More money for equity holders (YOU!)")

print("\\n✅ Operating model complete!")
```

This is the cash engine that pays down your debt!

Let's Build: Step 5 - Debt Schedule (THE CRITICAL PART!)

Create a new file: `05_debt_schedule.py`

```
"""
Step 5: Debt Schedule - THE MOST IMPORTANT PART OF AN LBO!

This is where the magic happens:
- Company generates Free Cash Flow
- FCF pays down debt
- Less debt = More equity value at exit!

This is WHY leverage works in PE.
"""

import pandas as pd

print("STEP 5: DEBT SCHEDULE")
print("=" * 70)
print("This is THE HEART of an LBO model!")
print("=" * 70)

# Starting debt
senior_debt_start = 200.0
sub_debt_start = 40.0
total_debt_start = senior_debt_start + sub_debt_start

# Interest rates
senior_rate = 0.05  # 5.0% interest
sub_rate = 0.09     # 9.0% interest (riskier, higher rate)

# Free cash flow (from Step 4)
years = list(range(2025, 2030))
fcf = [40.8, 44.0, 47.7, 51.2, 54.7]  # Simplified from Step 4

print(f"\\nStarting Debt:")
print(f"  Senior Debt:       ${senior_debt_start:.0f}M @
{senior_rate*100:.1f}%")
print(f"  Subordinated Debt: ${sub_debt_start:.0f}M @
{sub_rate*100:.1f}%")
print(f"  Total Debt:        ${total_debt_start:.0f}M")

# Build debt schedule
debt_schedule = []

senior_balance = senior_debt_start
sub_balance = sub_debt_start

for i, year in enumerate(years):
    # Calculate interest
    senior_interest = senior_balance * senior_rate
    sub_interest = sub_balance * sub_rate
    total_interest = senior_interest + sub_interest

    # Cash available for debt paydown
    cash_for_debt = fcf[i] - total_interest
```

```
    # Pay down senior debt first (it has priority!)
    if cash_for_debt > 0:
        senior_paydown = min(cash_for_debt, senior_balance)
        senior_balance -= senior_paydown

        # If senior is paid off, pay down sub debt
        remaining_cash = cash_for_debt - senior_paydown
        if remaining_cash > 0:
            sub_paydown = min(remaining_cash, sub_balance)
            sub_balance -= sub_paydown
        else:
            sub_paydown = 0
    else:
        senior_paydown = 0
        sub_paydown = 0

    total_balance = senior_balance + sub_balance

    debt_schedule.append({
        'Year': year,
        'FCF': fcf[i],
        'Senior_Interest': senior_interest,
        'Sub_Interest': sub_interest,
        'Total_Interest': total_interest,
        'Senior_Paydown': senior_paydown,
        'Sub_Paydown': sub_paydown,
        'Senior_Balance': senior_balance,
        'Sub_Balance': sub_balance,
        'Total_Debt': total_balance
    })

debt_df = pd.DataFrame(debt_schedule)

print("\\nDebt Schedule ($M):")
print(debt_df[['Year', 'FCF', 'Total_Interest', 'Senior_Paydown',
'Sub_Paydown', 'Total_Debt']].round(1))

# Summary
print("\\n" + "=" * 70)
print("DEBT PAYDOWN SUMMARY:")
print("=" * 70)

total_debt_paid = total_debt_start - debt_df['Total_Debt'].iloc[-1]
total_interest_paid = debt_df['Total_Interest'].sum()

print(f"Starting Total Debt:     ${total_debt_start:.0f}M")
print(f"Ending Total Debt:       ${debt_df['Total_Debt'].iloc[-1]:.0f}M")
print(f"\\nTotal Debt Paid Down:    ${total_debt_paid:.0f}M 🎉")
print(f"Total Interest Paid:     ${total_interest_paid:.0f}M")
print(f"\\nDebt Paydown %:          {(total_debt_paid / total_debt_start)
* 100:.1f}%")

# The magic of leverage
print("\\n" + "-" * 70)
```

```
print("THE MAGIC OF LEVERAGE:")
print("-" * 70)
print(f"You started with ${total_debt_start:.0f}M of OTHER PEOPLE'S MONEY")
print(f"The COMPANY paid down ${total_debt_paid:.0f}M of it")
print(f"You didn't use ANY of your equity! 💰")
print(f"\nAt exit, less debt = MORE equity value for YOU!")

print("\n✅ Debt schedule complete!")
print("💡 This is why PE firms love leverage - debt paydown is FREE money!")
```

THIS IS THE MAGIC! The company pays down debt with its own cash flow!

Let's Build: Step 6 - Exit Valuation

Create a new file: `06_exit_valuation.py`

```
"""
Step 6: Exit Valuation - What Can We Sell For?

After 5 years, you sell the company.
Exit Value = Exit EBITDA × Exit Multiple

PE firms make money from:
1. EBITDA growth (operating improvements)
2. Multiple expansion (selling at higher multiple than entry)
3. Debt paydown (more equity value!)
"""

import pandas as pd

print("STEP 6: EXIT VALUATION")
print("=" * 70)

# Exit year metrics (Year 5)
exit_revenue = 403.0  # From projections
exit_ebitda = 80.6    # 20% margin

# Entry vs Exit
entry_ebitda = 50.0
entry_multiple = 8.0
exit_multiple = 10.0  # Multiple expansion!

print("Entry vs Exit:")
print("-" * 70)
print(f"Entry EBITDA:     ${entry_ebitda:.0f}M @ {entry_multiple:.1f}x = ${entry_ebitda * entry_multiple:.0f}M")
print(f"Exit EBITDA:      ${exit_ebitda:.0f}M @ {exit_multiple:.1f}x = ${exit_ebitda * exit_multiple:.0f}M")
```

```
# Calculate exit value
exit_enterprise_value = exit_ebitda * exit_multiple

print("\\n" + "=" * 70)
print("EXIT VALUATION:")
print("=" * 70)
print(f"Exit EBITDA:                $ {exit_ebitda:.0f}M")
print(f"Exit Multiple:              {exit_multiple:.1f}x")
print(f"Exit Enterprise Value:      ${exit_enterprise_value:.0f}M")

# Value creation breakdown
print("\\n" + "-" * 70)
print("VALUE CREATION SOURCES:")
print("-" * 70)

# 1. EBITDA growth
ebitda_growth = exit_ebitda - entry_ebitda
value_from_ebitda_growth = ebitda_growth * entry_multiple  # At same
multiple

print(f"1. EBITDA Growth:")
print(f"   ${entry_ebitda:.0f}M → ${exit_ebitda:.0f}M
(+${ebitda_growth:.0f}M)")
print(f"   Value created: ${value_from_ebitda_growth:.0f}M @
{entry_multiple:.1f}x")

# 2. Multiple expansion
multiple_expansion = exit_multiple - entry_multiple
value_from_multiple = exit_ebitda * multiple_expansion

print(f"\\n2. Multiple Expansion:")
print(f"   {entry_multiple:.1f}x → {exit_multiple:.1f}x (+
{multiple_expansion:.1f}x)")
print(f"   Value created: ${value_from_multiple:.0f}M on
${exit_ebitda:.0f}M EBITDA")

# Total value created
entry_ev = entry_ebitda * entry_multiple
total_value_created = exit_enterprise_value - entry_ev

print(f"\\n" + "=" * 70)
print(f"TOTAL VALUE CREATED: ${total_value_created:.0f}M")
print(f"=" * 70)
print(f"  From EBITDA Growth:     ${value_from_ebitda_growth:.0f}M
({(value_from_ebitda_growth/total_value_created)*100:.0f}%)")
print(f"  From Multiple Expansion: ${value_from_multiple:.0f}M
({(value_from_multiple/total_value_created)*100:.0f}%)")

# Debt consideration
remaining_debt = 98.8  # From debt schedule
exit_equity_value = exit_enterprise_value - remaining_debt

print("\\n" + "-" * 70)
```

```
print("BRIDGE TO EQUITY VALUE:")
print("-" * 70)
print(f"Exit Enterprise Value:    ${exit_enterprise_value:.0f}M")
print(f"Less: Remaining Debt:     ${remaining_debt:.0f}M")
print(f"Exit Equity Value:        ${exit_equity_value:.0f}M")

print("\\n✅ Exit valuation complete!")
print("💡 PE firms target 2-3x multiple expansion from entry to exit")
```

The exit is where you realize all your hard work!

Let's Build: Step 7 - Returns Calculation (The Moment of Truth!)

Create a new file: `07_returns_calculation.py`

```
"""
Step 7: Returns Calculation - Did We Make Money?

Calculate IRR (Internal Rate of Return) and MOIC (Multiple on Invested
Capital)

These are THE TWO METRICS every PE investor cares about!
"""

import pandas as pd
import numpy as np

print("STEP 7: RETURNS CALCULATION")
print("=" * 70)
print("The moment of truth - did we make money?? 💰 ")
print("=" * 70)

# Investment (from Sources & Uses)
equity_invested = 180.0  # $180M equity check

# Exit (from Exit Valuation)
exit_equity_value = 707.2  # $707M equity at exit
holding_period = 5  # years

# MOIC (Multiple on Invested Capital)
moic = exit_equity_value / equity_invested

print("\\nINVESTMENT:")
print("-" * 70)
print(f"Equity Invested:    ${equity_invested:.0f}M")
print(f"Exit Equity Value:  ${exit_equity_value:.0f}M")
print(f"Holding Period:     {holding_period} years")

print("\\n" + "=" * 70)
print(f"MOIC: {moic:.2f}x")
```

```
print("=" * 70)

# Interpret MOIC
if moic >= 3.0:
    rating = "🔥 HOME RUN!"
elif moic >= 2.5:
    rating = "🚀 EXCELLENT!"
elif moic >= 2.0:
    rating = "✅ GOOD"
elif moic >= 1.5:
    rating = "🟡 OKAY"
else:
    rating = "🔴 POOR"

print(f"Rating: {rating}")

# IRR (Internal Rate of Return)
# IRR is the discount rate where NPV = 0
# Simplified: IRR ≈ (MOIC ^ (1/years)) - 1

irr = (moic ** (1/holding_period)) - 1

print("\\n" + "=" * 70)
print(f"IRR: {irr * 100:.1f}%")
print("=" * 70)

# Interpret IRR
if irr >= 0.30:
    rating = "🔥 EXCEPTIONAL! Top quartile return"
elif irr >= 0.25:
    rating = "🚀 HOME RUN! Partners very happy"
elif irr >= 0.20:
    rating = "✅ STRONG. Do the deal!"
elif irr >= 0.15:
    rating = "🟡 ACCEPTABLE. Borderline"
else:
    rating = "🔴 BELOW HURDLE. Pass."

print(f"Rating: {rating}")

# Cash-on-cash analysis
print("\\n" + "-" * 70)
print("CASH-ON-CASH ANALYSIS:")
print("-" * 70)

profit = exit_equity_value - equity_invested
print(f"Profit:              ${profit:.0f}M")
print(f"Return on Equity:    {(profit / equity_invested) * 100:.0f}%")
print(f"\\nYou invested:         ${equity_invested:.0f}M")
print(f"You got back:        ${exit_equity_value:.0f}M")
print(f"You made:            ${profit:.0f}M in {holding_period} years! 💰 💰
💰 ")

# Annualized return
```

```
annualized = profit / holding_period
print(f"\\nAnnualized profit:  ${annualized:.0f}M per year")

# What does this mean?
print("\\n" + "=" * 70)
print("WHAT DOES THIS MEAN?")
print("=" * 70)
print(f"If you invested ${equity_invested:.0f}M in a typical S&P 500 index fund:")
print(f"  @ 10% annual return for 5 years")
print(f"  You'd have: ${equity_invested * (1.10 ** 5):.0f}M")
print(f"\\nWith this LBO, you have: ${exit_equity_value:.0f}M")
print(f"\\nYou beat the market by ${exit_equity_value - equity_invested * (1.10 ** 5):.0f}M!")
print(f"That's why PE firms charge 20% carried interest! 🚀")

print("\\n✅ Returns calculation complete!")
print("💡 Always remember: 20%+ IRR is the PE industry standard")
```

THIS IS IT! Did we hit our target returns?

Let's Build: Step 8 - Sensitivity Analysis

Create a new file: `08_sensitivity_analysis.py`

```
"""
Step 8: Sensitivity Analysis - Testing Different Scenarios

What if things don't go as planned?
Test how returns change with different assumptions!

This is what PE Partners ask: "What if revenue growth is slower?"
"""

import pandas as pd
import numpy as np

print("STEP 8: SENSITIVITY ANALYSIS")
print("=" * 70)

# Base case
equity_invested = 180.0
entry_multiple = 8.0
entry_ebitda = 50.0

def calculate_returns(exit_ebitda, exit_multiple, remaining_debt):
    """Calculate MOIC and IRR for given scenario"""
    exit_ev = exit_ebitda * exit_multiple
    exit_equity = exit_ev - remaining_debt
    moic = exit_equity / equity_invested
```

```
    irr = (moic ** 0.2) - 1  # 5-year hold
    return moic, irr

# SENSITIVITY 1: Exit Multiple vs Exit EBITDA
print("\\nSENSITIVITY #1: EXIT MULTIPLE VS EXIT EBITDA")
print("=" * 70)

exit_ebitda_range = [70, 75, 80, 85, 90]  # Different EBITDA scenarios
exit_multiple_range = [8.0, 9.0, 10.0, 11.0, 12.0]  # Different multiples
remaining_debt = 98.8  # Assuming base case debt paydown

# Create sensitivity table for IRR
irr_table = []
for ebitda in exit_ebitda_range:
    row = []
    for multiple in exit_multiple_range:
        moic, irr = calculate_returns(ebitda, multiple, remaining_debt)
        row.append(irr * 100)  # Convert to percentage
    irr_table.append(row)

sensitivity_df = pd.DataFrame(
    irr_table,
    index=[f'${e}M' for e in exit_ebitda_range],
    columns=[f'{m:.1f}x' for m in exit_multiple_range]
)
sensitivity_df.index.name = 'Exit EBITDA'
sensitivity_df.columns.name = 'Exit Multiple →'

print("\\nIRR Sensitivity Table (%):")
print(sensitivity_df.round(1))

# Highlight key scenarios
print("\\n" + "-" * 70)
print("KEY SCENARIOS:")
print("-" * 70)

base_moic, base_irr = calculate_returns(80.6, 10.0, 98.8)
print(f"BASE CASE ($80M EBITDA, 10.0x):  {base_irr*100:.1f}% IRR, {base_moic:.2f}x MOIC ✅ ")

bull_moic, bull_irr = calculate_returns(90, 11.0, 80)  # Better ops, better exit, more debt paid
print(f"BULL CASE ($90M EBITDA, 11.0x):  {bull_irr*100:.1f}% IRR, {bull_moic:.2f}x MOIC 🚀 ")

bear_moic, bear_irr = calculate_returns(70, 8.0, 120)  # Worse ops, flat multiple, less debt paid
print(f"BEAR CASE ($70M EBITDA, 8.0x):   {bear_irr*100:.1f}% IRR, {bear_moic:.2f}x MOIC 🔴 ")

# SENSITIVITY 2: Revenue Growth Impact
print("\\n" + "=" * 70)
print("SENSITIVITY #2: REVENUE GROWTH IMPACT")
print("=" * 70)
```

```
base_revenue = 250.0
ebitda_margin = 0.20

print("\\nHow Revenue Growth Affects Returns:")
print("-" * 70)
print(f"{'CAGR':<10} {'Exit Revenue':<15} {'Exit EBITDA':<15} {'@ 10.0x Exit':<15} {'IRR':<10}")
print("-" * 70)

for cagr in [0.06, 0.08, 0.10, 0.12, 0.14]:
    exit_revenue = base_revenue * ((1 + cagr) ** 5)
    exit_ebitda = exit_revenue * ebitda_margin
    exit_ev = exit_ebitda * 10.0
    exit_equity = exit_ev - remaining_debt
    moic = exit_equity / equity_invested
    irr = (moic ** 0.2) - 1

    print(f"{cagr*100:>5.0f}%     ${exit_revenue:>10.0f}M      ${exit_ebitda:>10.0f}M      ${exit_ev:>10.0f}M      {irr*100:>5.1f}%")

# What matters most?
print("\\n" + "=" * 70)
print("WHAT MATTERS MOST?")
print("=" * 70)

print("\\nTesting +10% change in each variable:")
print("-" * 70)

# Base case
base_moic, base_irr = calculate_returns(80, 10.0, 98.8)

# Test EBITDA +10%
moic_ebitda, irr_ebitda = calculate_returns(88, 10.0, 98.8)
impact_ebitda = (irr_ebitda - base_irr) * 100

# Test Exit Multiple +10%
moic_multiple, irr_multiple = calculate_returns(80, 11.0, 98.8)
impact_multiple = (irr_multiple - base_irr) * 100

# Test Debt Paydown +10%
moic_debt, irr_debt = calculate_returns(80, 10.0, 88.9)
impact_debt = (irr_debt - base_irr) * 100

print(f"Exit EBITDA +10%:     IRR changes by {impact_ebitda:+.1f} percentage points")
print(f"Exit Multiple +10%:   IRR changes by {impact_multiple:+.1f} percentage points")
print(f"Debt Paydown +10%:    IRR changes by {impact_debt:+.1f} percentage points")

print("\\n💡 INSIGHT:")
if abs(impact_multiple) > abs(impact_ebitda):
    print("   Exit multiple has BIGGER impact than EBITDA growth!")
```

```
    print("   → Focus on buying at low multiple, selling at high multiple")
else:
    print("   EBITDA growth is KEY to returns!")
    print("   → Focus on operational improvements")

print("\\n✅ Sensitivity analysis complete!")
print("💡 Always run sensitivities - things rarely go exactly as planned!")
```

Test every scenario! This is what PE Partners review in Investment Committee.

The Complete LBO Class - `lbo_model.py`

The file `lbo_model.py` contains a **production-quality LBO class** that you can use for real deals!

Practice Exercises

Time to test your LBO skills! 🎯

Exercise 1: Quick LBO Returns Check

You're evaluating a new deal:

- Entry: $60M EBITDA @ 7.5x = $450M purchase price
- Financing: 40% equity ($180M), 60% debt ($270M)
- Projections: EBITDA grows to $90M in 5 years
- Exit: 9.0x EBITDA
- Assume $120M debt remaining at exit

Calculate:

1. Exit enterprise value
2. Exit equity value
3. MOIC
4. IRR

Is this a good deal? (Target: 20%+ IRR)

Exercise 2: Build a Complete LBO Model

Pick a real public company and model an LBO:

1. Find current EBITDA (use Yahoo Finance)
2. Assume you buy at 8.0x EBITDA
3. Finance with 40% equity, 60% debt
4. Project 5 years of growth (conservative!)
5. Build debt schedule
6. Assume exit at 9.0x EBITDA

7. Calculate returns

Compare to the company's current stock performance - would the LBO beat it?

Exercise 3: Sensitivity Analysis

For your Exercise 2 model, create:

1. 2-way sensitivity table (Exit EBITDA vs Exit Multiple)
2. Test revenue growth scenarios (5%, 10%, 15% CAGR)
3. Test different leverage levels (30%, 40%, 50%, 60% equity)
4. Which variable drives returns most?

Exercise 4: Deal Comparison

You have two deals to choose from (same $180M equity check):

Deal A: Mature company

- Entry: $50M EBITDA @ 7.0x
- Growth: 6% CAGR
- Exit: 7.5x (limited expansion)

Deal B: Growth company

- Entry: $35M EBITDA @ 9.0x
- Growth: 15% CAGR
- Exit: 11.0x (high multiple)

Model both deals and decide: Which has better returns? Which is less risky?

Solutions

Complete solutions available in `solutions.py`. Try the exercises first!

Key Takeaways

✅ **LBOs use leverage to amplify returns** (buy with debt, company pays it down) ✅ **Make money 3 ways:** EBITDA growth + Multiple expansion + Debt paydown ✅ **Target 20%+ IRR, 2.5-3.0x MOIC** in 5 years ✅ **Debt schedule is critical** - FCF pays down debt! ✅ **Always run sensitivities** - test bear/base/bull cases ✅ **Entry multiple matters** - buy low, sell high!

The LBO is THE signature investment strategy of private equity!

🎉 Congratulations!

You now understand how PE firms like Blackstone, KKR, and PE Club evaluate buyout opportunities!

Next Steps:

Continue to: `Module_06_MA_Analysis/01_MA_Overview.md`

Learn M&A analysis from the investment banking side!

Estimated Time: 5-6 hours **Difficulty:** ⭐⭐⭐⭐ Advanced **Prerequisites:** Modules 1-4 (especially Module 4 DCF)

Real-world impact: This is EXACTLY what you do at PE Club when evaluating buyout opportunities!

You're now ready for PE investing! 🚀 💰