

# Excel and Python TVM

Nick DeRobertis

March 3, 2021

## Overview

### The Problem

You work for a new startup that is trying to manufacture phones. You are tasked with building a model which will help determine how many machines to invest in and how much to spend on marketing. Each machine produces  $n_{output}$  phones per year. Each phone sells for  $\$p_{phone}$  and costs  $\$c_{phone}$  in variable costs to produce. After  $n_{life}$  years, the machine can no longer produce output, but may be scrapped for  $\$p_{scrap}$ . The machine will not be replaced, so you may end up with zero total output before your model time period ends. Equity investment is limited, so in each year you can spend  $c_{machine}$  to either buy a machine or buy advertisements. In the first year you must buy a machine. Any other machine purchases must be made one after another (advertising can only begin after machine buying is done). Demand for your phones starts at  $d_1$ . Each time you advertise, demand increases by  $g_d\%$ . The prevailing market interest rate is  $r$ .

### Notes

- You may limit your model to 20 years and a maximum of 5 machines if it is helpful.
- For simplicity, assume that  $c_{machine}$  is paid in every year, even after all machines have shut down.
- Ensure that you can change the inputs and the outputs change as expected.
- For simplicity, assume that fractional phones can be sold, you do not need to round the quantity transacted.

### The Model

#### Inputs

- $n_{output}$ : Number of phones per machine per year
- $n_{machines}$ : Number of machines purchased
- $n_{life}$ : Number of years for which the machine produces phones
- $p_{phone}$ : Price per phone
- $p_{scrap}$ : Scrap value of machine
- $c_{machine}$ : Price per machine or advertising year
- $c_{phone}$ : Variable cost per phone
- $d_1$ : Quantity of phones demanded in the first year
- $g_d$ : Percentage growth in demand for each advertisement
- $r$ : Interest rate earned on investments

## Outputs

- Cash flows in each year, up to 20 years
- PV of cash flows, years 1 - 20

## Bonus Problem

It is unrealistic to assume that price and demand are unrelated. To extend the model, we can introduce a relationship between price and demand, given by the following equation:

$$d_1 = d_c - Ep_{phone} \quad (1)$$

- $E$ : Price elasticity of demand
- $d_c$ : Demand constant

For elasticities and constants  $[(E = 500, d_c = 900000), (E = 200, d_c = 500000), (E = 100, d_c = 300000)]$  (3 total cases), and taking the other model inputs in the Check your Work section, determine the optimal price for each elasticity, that is the price which maximizes the NPV.

## Notes

- $d_1$  is no longer an input, but an output.
- This bonus requires optimization, which we have not yet covered in class.
- In Excel, you can use Solver.
- In Python, the `scipy` package provides optimization tools. You will probably want to use:
  - [scipy.optimize.minimize\\_scalar](#)
  - You will need to write a function which accepts price and returns NPV, with other model inputs fixed.
    - \* Depending on how you set this up, [functools.partial](#) may be helpful for this.
  - It will actually need to return negative NPV, as the optimizer only minimizes, but we want maximum NPV.
  - No answers to check your work are given for this bonus. The Check your Work section only applies to without the bonus.

## Excel Exercise

You must start from "Project 1 Template.xlsx". Ensure that you reference all inputs from the Inputs/Outputs tab. Also ensure that all outputs are referenced back to the Inputs/Outputs tab. Do not change any locations of the inputs or outputs. The final submission is your Excel workbook.

## Python Exercise

You must start from "Project 1 Template.ipynb". I should be able to run all the cells and get the output of your model at the bottom. You should not change the name of the `ModelInputs` class or the `model_data` variable. You need to define `cash_flows` as your output cash flows (numbers, not formatted), and `npv` as your NPV (number, not formatted). When you show your final outputs in the notebook, then they should be formatted.

## Grading

Grading Breakdown	
Category	Percentage
Model Accuracy	60%
Model Readability	20%
Model Formatting	10%
Following the Template	10%
Bonus	5%
Total Possible	105%

## Check your Work

If you pass the following inputs (to the basic model, not bonus model):

- $n_{output}$ : 100,000
- $p_{scrap}$ : \$50,000
- $p_{phone}$ : \$500
- $c_{machine}$ : \$1,000,000
- $c_{phone}$ : \$250
- $n_{life}$ : 10
- $n_{machines}$ : 5
- $d_1$ : 100,000
- $g_d$ : 20%
- $r$ : 5%

You should get the following result:

Cash Flows:

Year 1: \$24,000,000  
Year 2: \$24,000,000  
Year 3: \$24,000,000  
Year 4: \$24,000,000  
Year 5: \$24,000,000  
Year 6: \$29,000,000  
Year 7: \$35,000,000  
Year 8: \$42,200,000  
Year 9: \$50,840,000  
Year 10: \$61,208,000  
Year 11: \$73,699,600  
Year 12: \$74,050,000  
Year 13: \$49,050,000  
Year 14: \$24,050,000  
Year 15: \$-950,000  
Year 16: \$-1,000,000  
Year 17: \$-1,000,000  
Year 18: \$-1,000,000  
Year 19: \$-1,000,000  
Year 20: \$-1,000,000  
NPV: \$369,276,542