

# CS 458/558 - Problem Set 2 - DRAFT

**Assigned:** Wednesday September 16

**Deadline:** Wednesday October 7 (new extension), 11:59pm

## Capital Budgeting and Net Present Value

In the lecture notes, we discussed the following case.

As an executive at Ford, I need to build a new auto plant. The options are a \$40 million plant in Ohio that can produce 1000 cars a month with union labor, with an expected lifetime of 12 years and a \$20 million dollar plant in South Carolina that can produce 500 cars a month with non-union labor and tax incentives and an expected lifetime of 10 years. Which is best?

This was insufficient to solve the problem. We provided additional data.

We can assume in each case that it takes two years to build the factory (which is included in the lifetime of the plant), and the cost is split evenly over those two years. Each car produced generates \$10,000 in profit, excluding production costs. Annual costs are \$6500 per car for union workers in \$4000 per car for non union workers. The discount rate is 5%.

For this problem set, we want you to implement a program in **either R or Python** which solves this problem, in three stages.

- **Stage 1: NPV.** Have a top level function, **decide** which takes a list of options. In our example, there are two choices, but **your function should handle any (positive) number of choices**. Each choice is represented as **a list of features** which are then passed to a second function `npv` which performs the calculation demonstrated in the notes. The `decide` function should then return which option (if any) is preferable.

- **Stage 2: Sensitivity Analysis.** Write a function `sensitivity` which varies the underlying assumptions, one at a time, and determines which of the assumptions, if any, would give you a better outcome. You might want `sensitivity` to call `decide`. For the purpose of this exercise, you should vary each numeric financial assumption by plus or minus 20%. For example, if the given discount rate is 5%, you should try 4% and 6%.

For the Ohio option, here are the possible scenario values:

Attribute	Low	High
cost	32,000,000	48,000,000
discount	.04	.06
costpercar	5,200	7,800
revenuepercar	8,000	12,000
monthlyoutput	800	1200

Thus, your `sensitivity` function should run a total of 10 different scenarios, each of which varies one of the parameters by plus or minus 20%.

- **Stage 3: Explanation to Stakeholders.** Now prepare to move out of your comfort zone. Your program is going to come up with a recommendation. For stage 3, you need to have your program explain or justify that recommendation. Write a function `explain` which takes one or two parameters. The first parameter is the output of either `decide` or `sensitivity`. The `explain` function will provide a justification for the decision. The optional second parameter is a stakeholder. If it is present, then the explanation needs to address the explicit concerns of that stakeholder. Here is a list of stakeholders:

- Stockholders
- Union members
- Ohio community
- South Carolina community

union member: 工会成员

I am not providing examples of explanations. That is up to you. I want you to think about this on your own. You can earn most of the credit for this assignment by doing a good job on stages 1 and 2. However, it is important for you to start thinking about the qualitative aspects of decision making. It is a recurring theme of this course.

## Data Structures

The choices are always assumed to be Ford automobile factories with the same set of stakeholders. The relevant parameters will be given by the following:

- option 1:

```
location: "OH",  
cost: 40000000,  
yearstocomplete: 2,  
lifetime: 12,  
discount: .05,  
union: TRUE,  
costpercar: 6500,  
revenuepercar: 10000,  
monthlyoutput: 1000,
```

- option 2:

```
location: "SC",  
cost: 20000000,  
yearstocomplete: 2,  
lifetime: 10,  
discount: .05,  
union: FALSE,  
costpercar: 4000,  
revenuepercar: 10000,  
monthlyoutput: 500,
```

Each of the numeric parameters are subject to the plus or minus 20% sensitivity analysis, (except for the years).

We stipulate an object oriented approach, with classes for decisions and options. You might also want classes for stakeholders and explanations. For example, the `decide` function can instantiate a decision object which has fields for multiple stakeholders and multiple options. The `npv` function could take an `option` parameter, which it updates and returns to `decide`.

Here is an example of the start of an object oriented approach in Python: [hw2.py](#) ([in codeskulptor](#))

Here is an example of the start of an object oriented approach in R: [hw2.R](#) (See R Reference Classes, e.g. <http://www.inside-r.org/r-doc/methods/ReferenceClasses>)