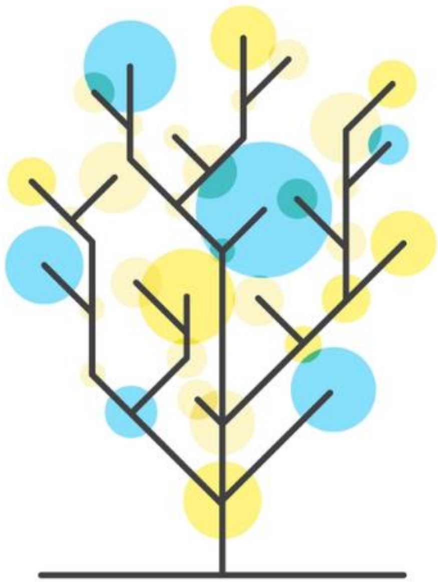
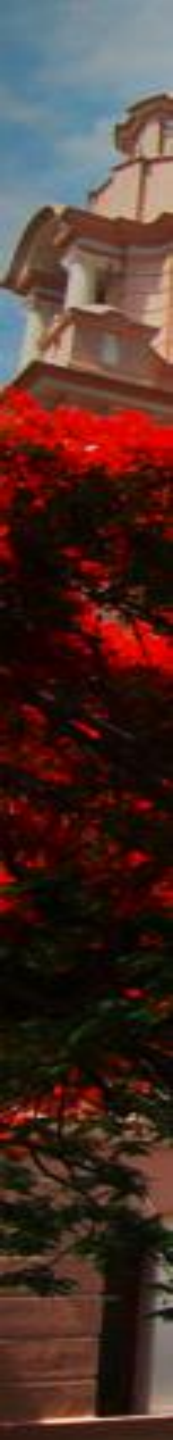
MSBA7003 Quantitative Analysis Methods

02 Decision Making under Risk

Agenda

- Decision making under risk
  - The six steps of decision making
  - Decision table
  - Expected value of information
- Decision tree
  - Sequential decision: sample collection
  - The use of historical data
- Use Python to Solve a Decision Tree
- Multiple-Stage Decision Making
  - Decision Policy
  - Revenue Management





Decision Making under Risk

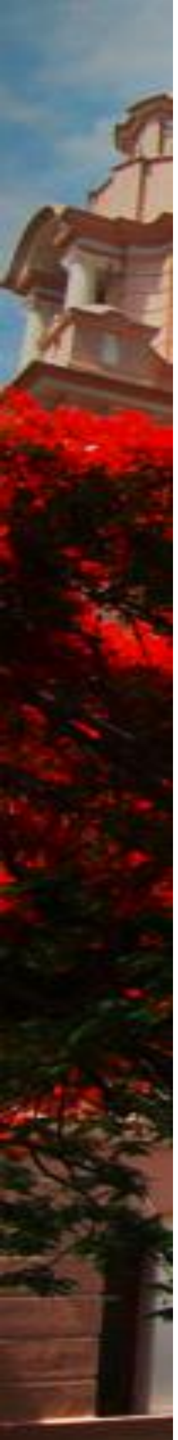
- Making a decision is basically making a choice.
  - Whether to pursue a graduate study
  - Whether to buy a stock and how much money to invest
  - Whether to expand the product line and how to expand

- Six Steps in Scientific Decision Making
  - Clearly define the problem (Goal to achieve)
  - List the possible alternatives
  - Identify the possible outcomes or states of nature
  - List the payoff of each alternative in each state of nature
  - Select one of the decision theory models
  - Apply the model and make your decision

Thompson Lumber Company

- John Thompson, the founder and president, makes the decision on whether to expand his product line by manufacturing and marketing a new product (a wooden shack).

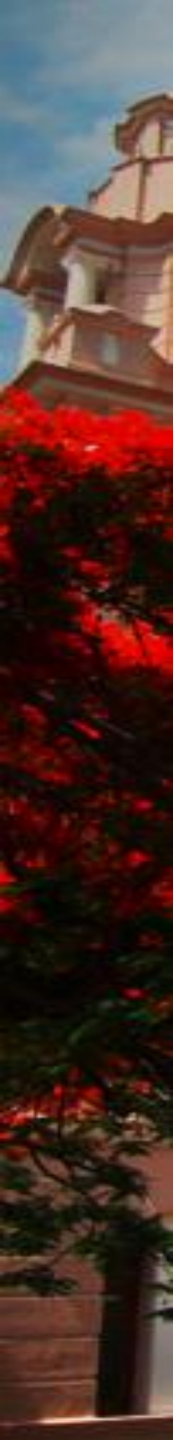

- There are three options: (1) construct a large plant with a cost of \$180,000, (2) construct a small plant with a cost of \$20,000, and (3) do nothing.
- In a favorable market, the gross profit will be \$380,000 with a large plant and \$120,000 with a small plant; in an unfavorable market, the gross profit will always be \$0.



Decision Making under Risk

ALTERNATIVE	STATE OF NATURE		Expected Monetary Value (EMV, $)
	FAVORABLE MARKET (profit in $)	UNFAVORABLE MARKET (profit in $)	
Construct a large plant	200,000	–180,000	–9,000
Construct a small plant	100,000	–20,000	34,000
Do nothing	0	0	0
Probability	0.45	0.55	

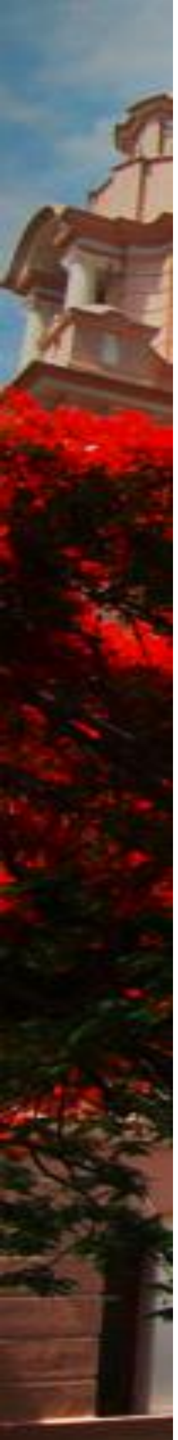
If John maximize the EMV, then he should choose to construct a small plant.



Decision Making under Risk

- Scientific Marketing, Inc. (S.M.) offers analysis that will provide certainty about market conditions. S.M. would charge $65,000 for the information. Should John buy the information?

- To make this decision, John has to evaluate the expected value of perfect information (EVPI) by computing the expected value with perfect information (EVwPI) and the best EMV under risk.
- **EVPI = EVwPI – Best EMV**

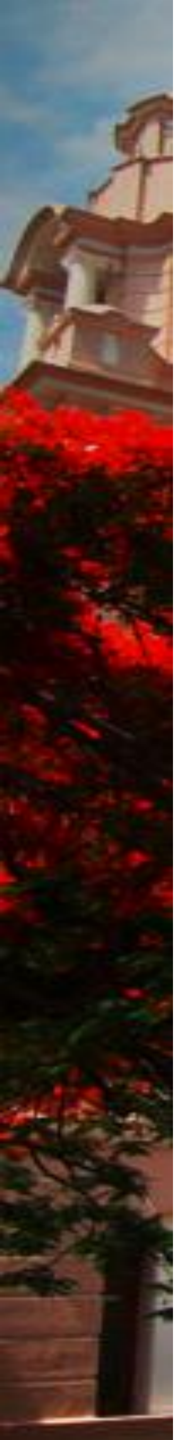


Decision Making under Risk

- EVwPI = $\sum(\text{best payoff in state } i) \cdot (\text{probability of state } i)$

	STATE OF NATURE		
ALTERNATIVE	FAVORABLE MARKET (profit in \$)	UNFAVORABLE MARKET (profit in \$)	Expected Monetary Value (EMV, \$)
Construct a large plant	200,000	–180,000	–9,000
Construct a small plant	100,000	–20,000	34,000
Do nothing	0	0	0
Best payoff	200,000	0	90,000
Probability	0.45	0.55	EVwPI

EVPI = \$90,000 – \$34,000 = \$56,000 < \$65,000. **Don't buy!**



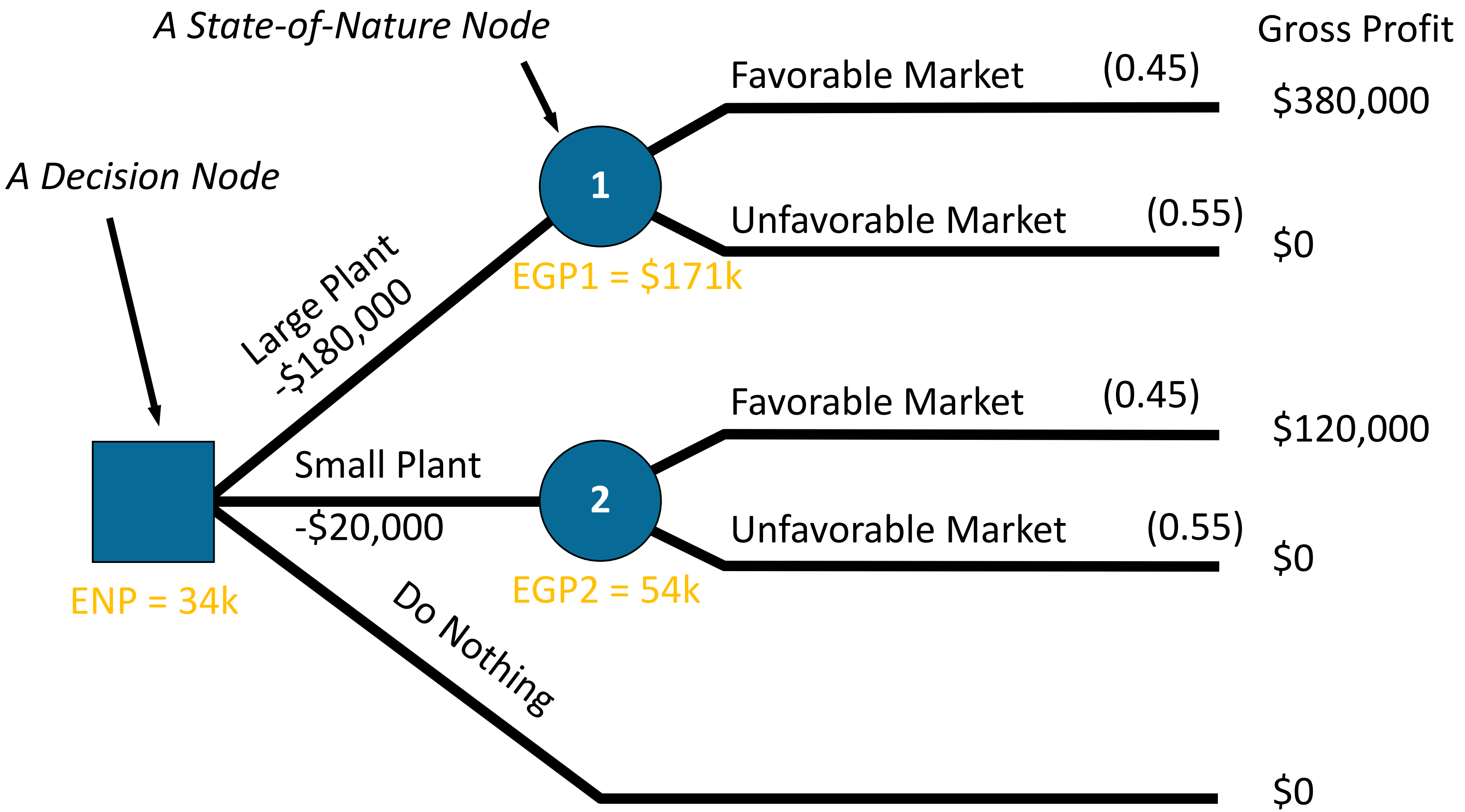
Decision Tree

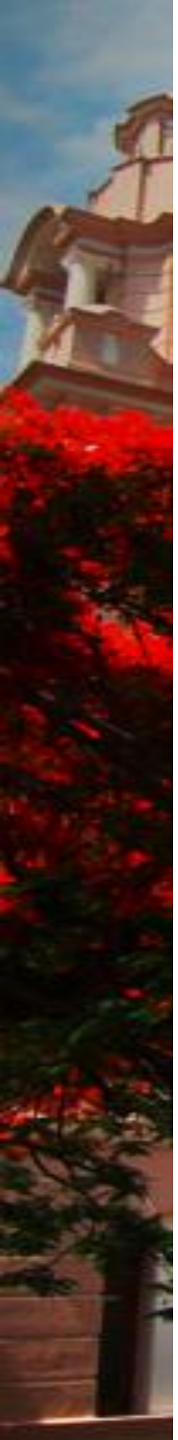
- Any problem that can be presented in a decision table can also be graphically illustrated in a decision tree.

- Advantages of decision tree:
  - Incorporate sequential decision making
  - Incorporate different states and probabilities for different options

- Any decision tree has two types of node:
  - *Decision node*, from which one of several options may be chosen
  - *State-of-nature node*, out of which one state of nature will occur

Thompson's Decision Tree





Decision Tree with Sample Information

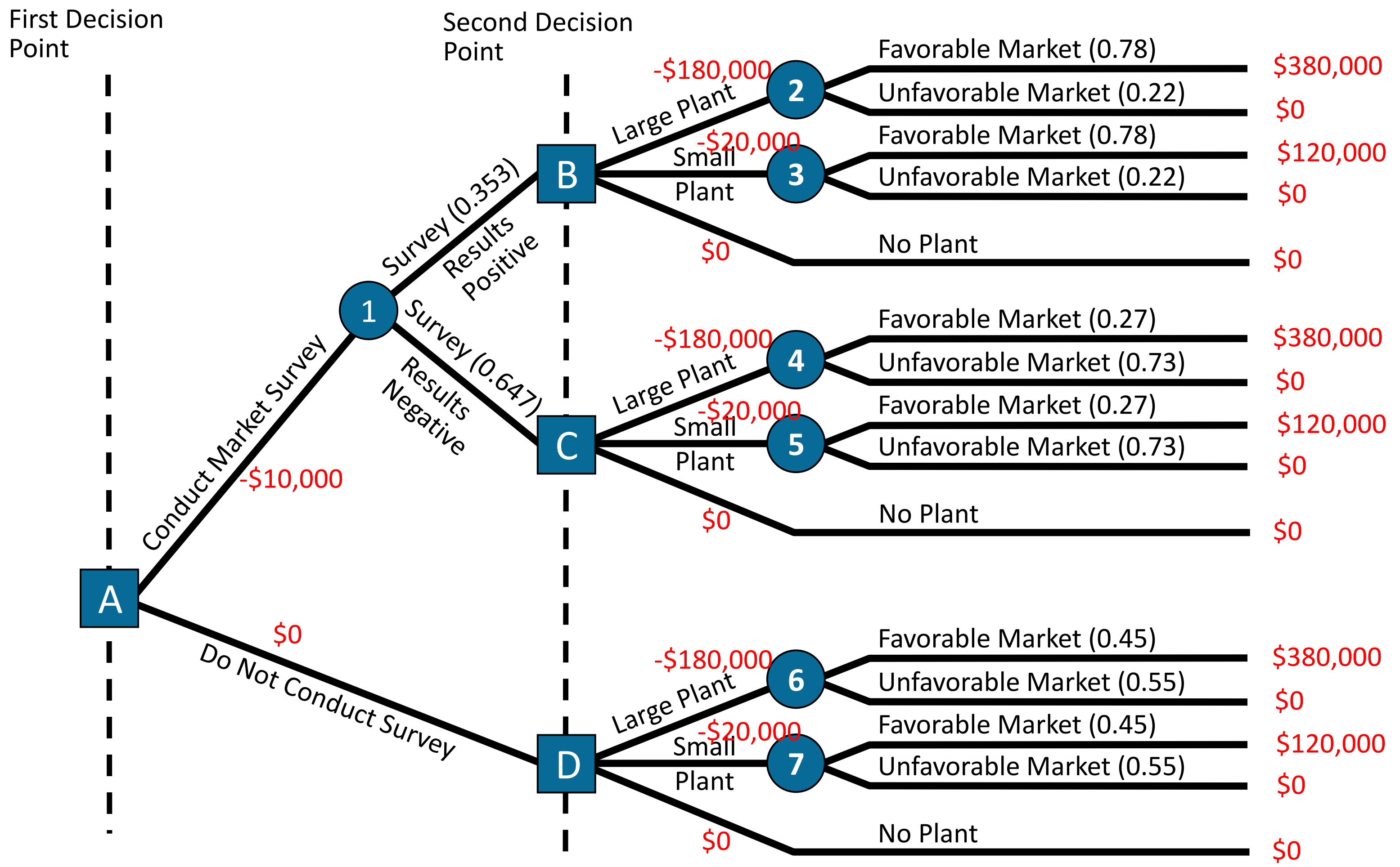
- Before deciding on building a new plant, John has the option of hiring ABC, Inc. to conduct a market survey, at a cost of $10,000.

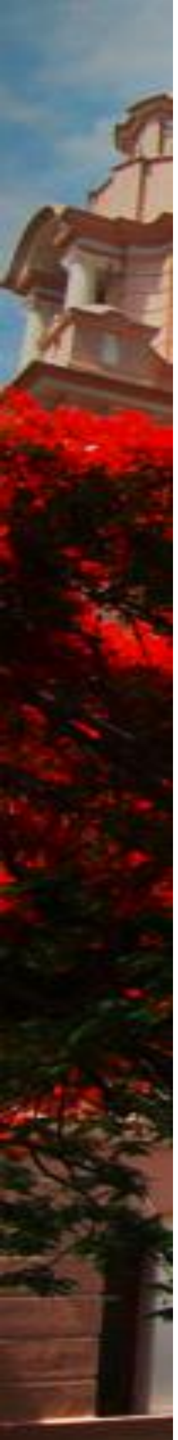
- Conditional on a positive survey result, the market will be favorable with probability 0.78 and unfavorable with probability 0.22.

- Conditional on a negative survey result, the market will be favorable with probability 0.27 and unfavorable with probability 0.73.

- P(Positive result) = 0.353; P(Negative result) = 0.647

Decision Tree with Sample Information





Decision Tree with Sample Information

- EGP = expected gross payoff going forward; ENP = expected gross net payoff going forward.
- Given favorable survey results,

EGP(node 2) = (0.78)($380,000) + (0.22)($0) = $296,400

EGP(node 3) = (0.78)($120,000) + (0.22)($0) = $93,600

EGP for no plant = $0

ENP(node B) = max($296,400 – $180,000, $93,600 – $20,000, $0) = $116,400

- Given negative survey results,

EGP(node 4) = (0.27)($380,000) + (0.73)($0) = $102,600

EGP(node 5) = (0.27)($120,000) + (0.73)($0) = $32,400

EGP for no plant = $0

ENP(node C) = max($102,600 – $180,000, $32,400 – $20,000, $0) = $12,400

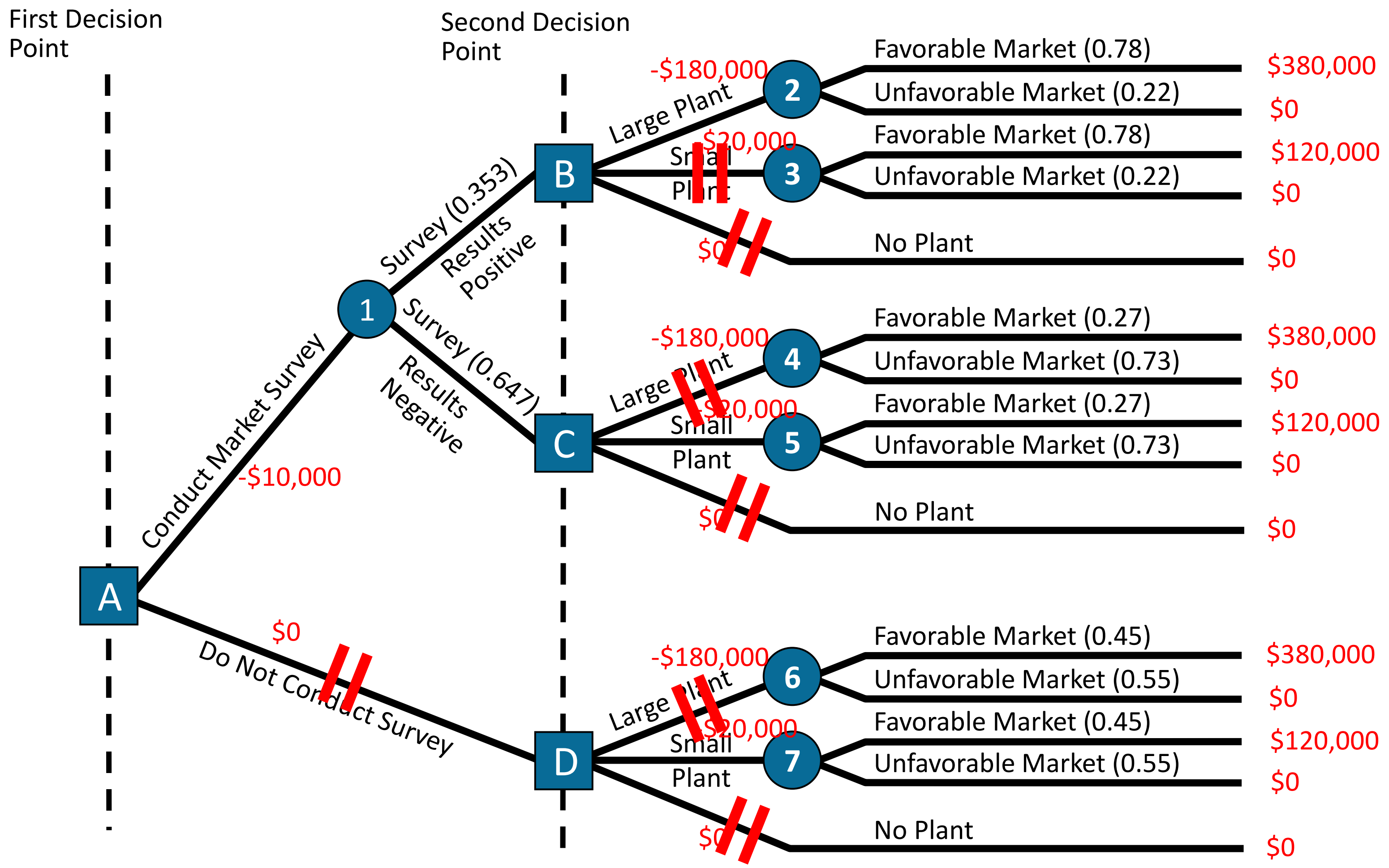
- For the first decision,

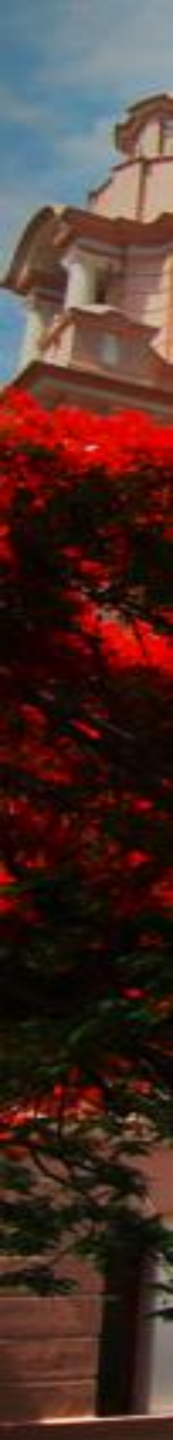
EGP(node 1) = (0.353)($116,400) + (0.647)(12,400) = $49,112

EGP(node D) = $34,000

ENP(node A) = max($49,112 – $10,000, $34,000) = $39,112

Decision Tree with Sample Information



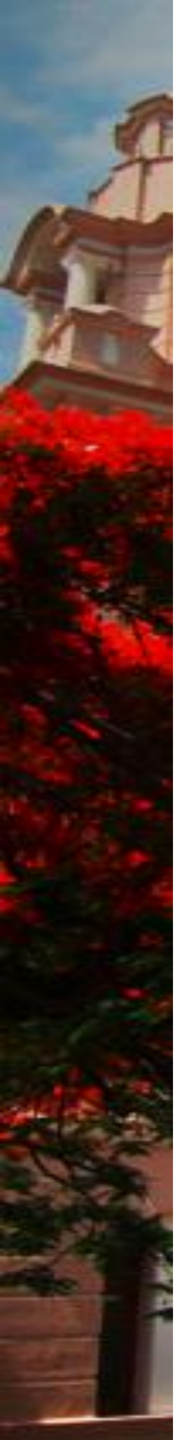


Use of Historical Data

- Suppose we do not know P(Pos), P(Neg), P(Fav|Pos), or P(Unf|Neg). Instead, we have a prior belief that P(Fav) = 0.8, and historical data from ABC, Inc.:

	Positive	Negative	Total
Favorable	35	15	50
Unfavorable	20	30	50
Total	55	45	

- Then P(Pos)=? P(Neg)=? P(Fav|Pos)=? P(Unf|Neg)=?
- Should John hire ABC to conduct the survey?



Use of Historical Data

- Prior belief is developed from private information and thus is usually useful.
- Historical data is obtained from other products, not the current one.
- The useful information we can obtain from data is the predictive power of ABC's survey.

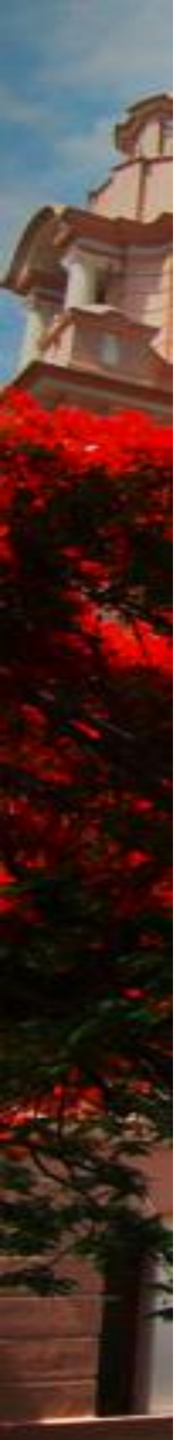
- How to measure the predictive power?
- Think about the HIV example:
- P(positive test result | infected) & P(negative test result | not infected).
- Note that P(infected | positive test result) heavily depends on the prior belief (prevalence rate).
- Therefore, we need P(Pos | Fav) and P(Neg | Unf).

Use of Historical Data

- For the current market:

Joint Prob.	Positive	Negative	Marginal
Favorable	0.8*35/50	0.8*15/50	0.8
Unfavorable	0.2*20/50	0.2*30/50	0.2
Marginal	14/25+2/25	6/25+3/25	

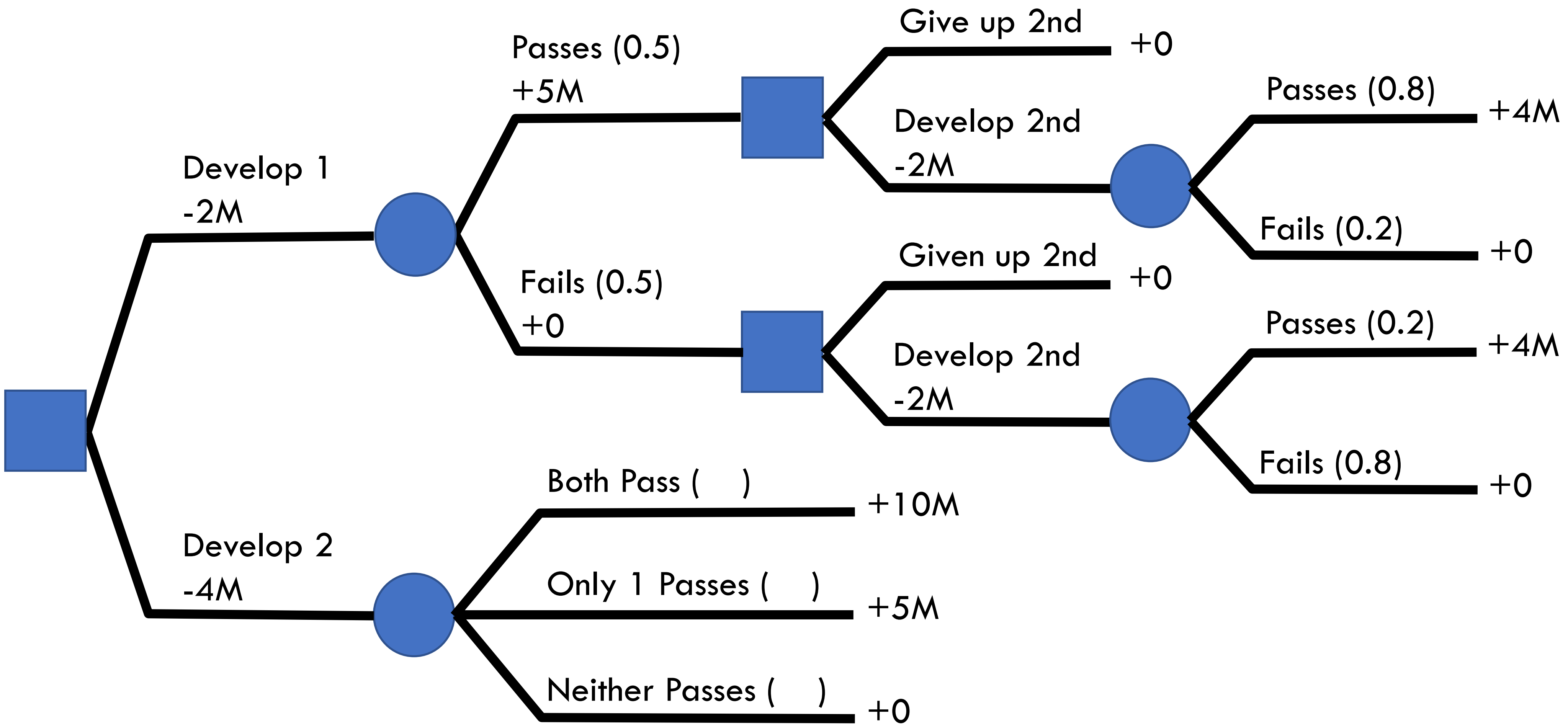
- P(Fav|Pos) = 14/25/(16/25) = 7/8
- P(Unf|Neg) = 3/25/(9/25) = 1/3
- The prior belief will be left unused if we calculate P(Fav|Pos) from the historical data.
- The final decision is left as a homework.

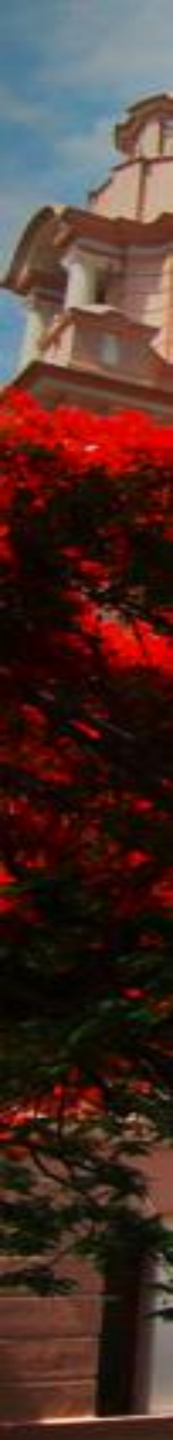


In-Class Exercises

- Fivecent is deciding whether to invest in two mobile games of similar themes but targeting different game devices and population.
- The initial development costs are both RMB 2 millions. The total expected revenue net of subsequent costs is RMB 5 millions (gross profit) for each game.
- Before launching a game, Fivecent must send the game to the government for approval. The prior belief is that the chance of getting approval is 50% for either game. If one is approved, the chance of getting approval is 80% for the other one. If one is rejected, the chance of being rejected is 80% for the other one.
- Should Fivecent develop the games sequentially or simultaneously? If games are to be developed sequentially, the delayed game will lose 20% of the gross profit.

In-Class Exercises





Use Python to Solve a Decision Tree

- Build the tree as a table (a Python dictionary)

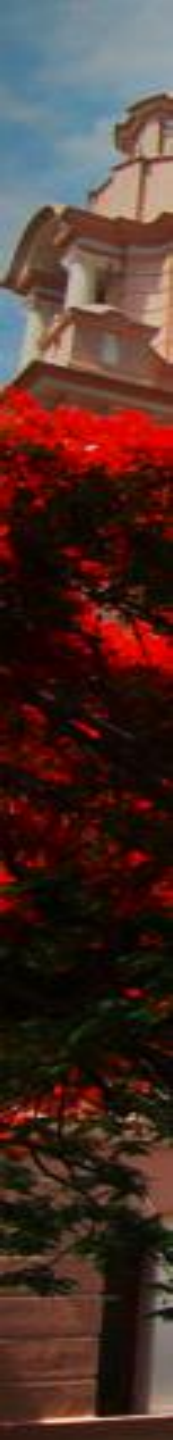
Index	Parent	Child	Note	Type	Cost	Probability	Cost-to-go
0	-	1,2	Root	D	-	-	0
1	0	3,4	Develop 1	S	2	-	0
2	0	5,6,7	Develop 2	S	4	-	0
3	1	8,9	1st passes	D	-5	0.5	0
4	1	10,11	1st fails	D	0	0.5	0
5	2	-	Both pass	L	-10	0.4	0
6	2	-	One passes	L	-5	0.2	0
7	2	-	Both fail	L	0	0.4	0
8	3	-	Give up 2nd	L	0	-	0
…	…	…	…	…	…	…	…

Use Python to Solve a Decision Tree

```
tree = {
    0:{'Parent':[],'Child':[1,2],'Note':'Root','Type':'D','Cost':0,'CTG':0},
    1:{'Parent':[0],'Child':[3,4],'Note':'Develop_1','Type':'S','Cost':2,'CTG':0},
    2:{'Parent':[0],'Child':[5,6,7],'Note':'Develop_2','Type':'S','Cost':4,'CTG':0},
    3:{'Parent':[1],'Child':[8,9],'Note':'1st_passes','Type':'D','Cost':-5,'Probability':0.5,'CTG':0},
    4:{'Parent':[1],'Child':[10,11],'Note':'1st_fails','Type':'D','Cost':0,'Probability':0.5,'CTG':0},
```

- Solve the tree through recursion

```
def cost_to_go(tree, node):
    if tree[node]['Type']=='L':
        tree[node]['CTG'] = 0
    else:
        if tree[node]['Type']=='D':
            child_CCTG = []
            for i in tree[node]['Child']:
                child_CCTG.append(tree[i]['Cost'] + cost_to_go(tree, i))
            tree[node]['CTG'] = min(child_CCTG)
        else:
            E_CCTG = 0
            for i in tree[node]['Child']:
                E_CCTG = E_CCTG + tree[i]['Probability']*(tree[i]['Cost']+cost_to_go(tree, i))
            tree[node]['CTG'] = E_CCTG
    return tree[node]['CTG']
```

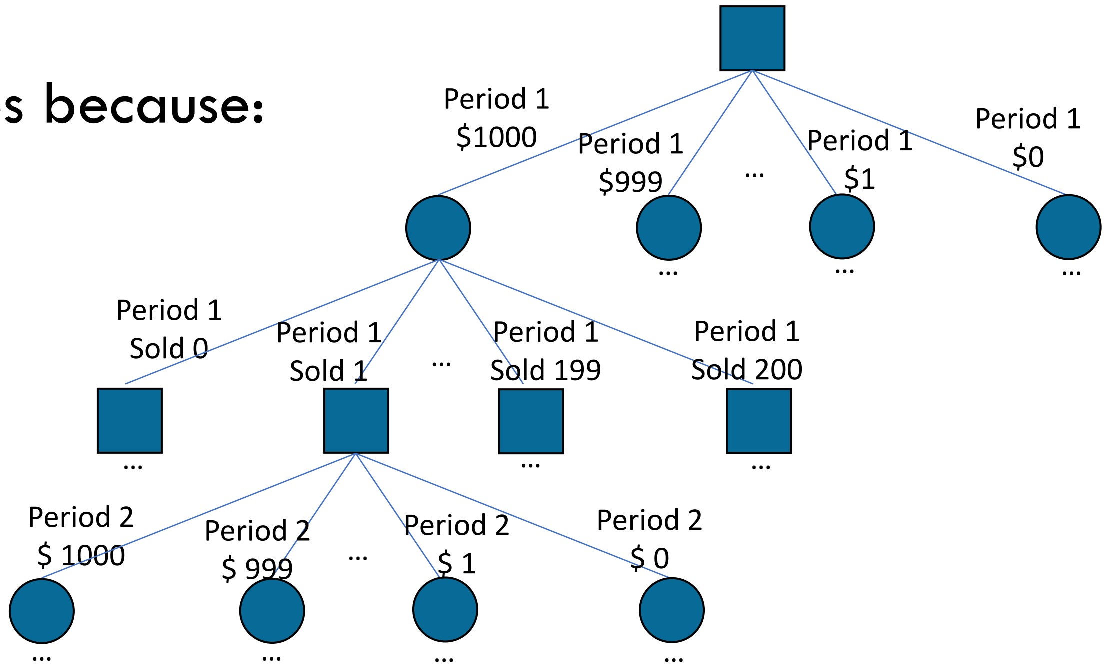


Multiple-Stage Decision Making

- When there are decisions to be made in multiple stages, early decisions can affect the *choice set* and *information set* of subsequent ones. Hence, we cannot just focus on the immediate payoff.
- The ideal way to solve a multiple-stage problem is through backward induction:
  - In the last stage, solve for the best decision and net payoff, given any previous decisions and the current state of the world;
  - Go to the preceding stage, solve for the best decision and net payoff, given any previous decisions and the current state of the world, and the gross payoff of each decision is calculated as the sum of the immediate payoff (if any) and the best net payoff in the subsequent stage after the decision;
  - Repeat the above procedures until we solve the first-stage problem.
- The theory: the optimal decision at a given node must be the true optimal one if subsequent decisions are all optimal.

Multiple-Stage Decision Making

- In the real business world, there are many complicated decision-making processes
  - Airline ticket pricing / Hotel room pricing
  - Game AI
- Backward induction is infeasible in these cases because:
  - The tree is too big (wide & deep)
  - It takes infinite amount of time to calculate
- Alternative solutions:
  - Train a decision-making policy (function)
    - Policy: (Stage, State) => Decision
  - Focus on a smaller part of the tree
    - Tree Search Algorithms

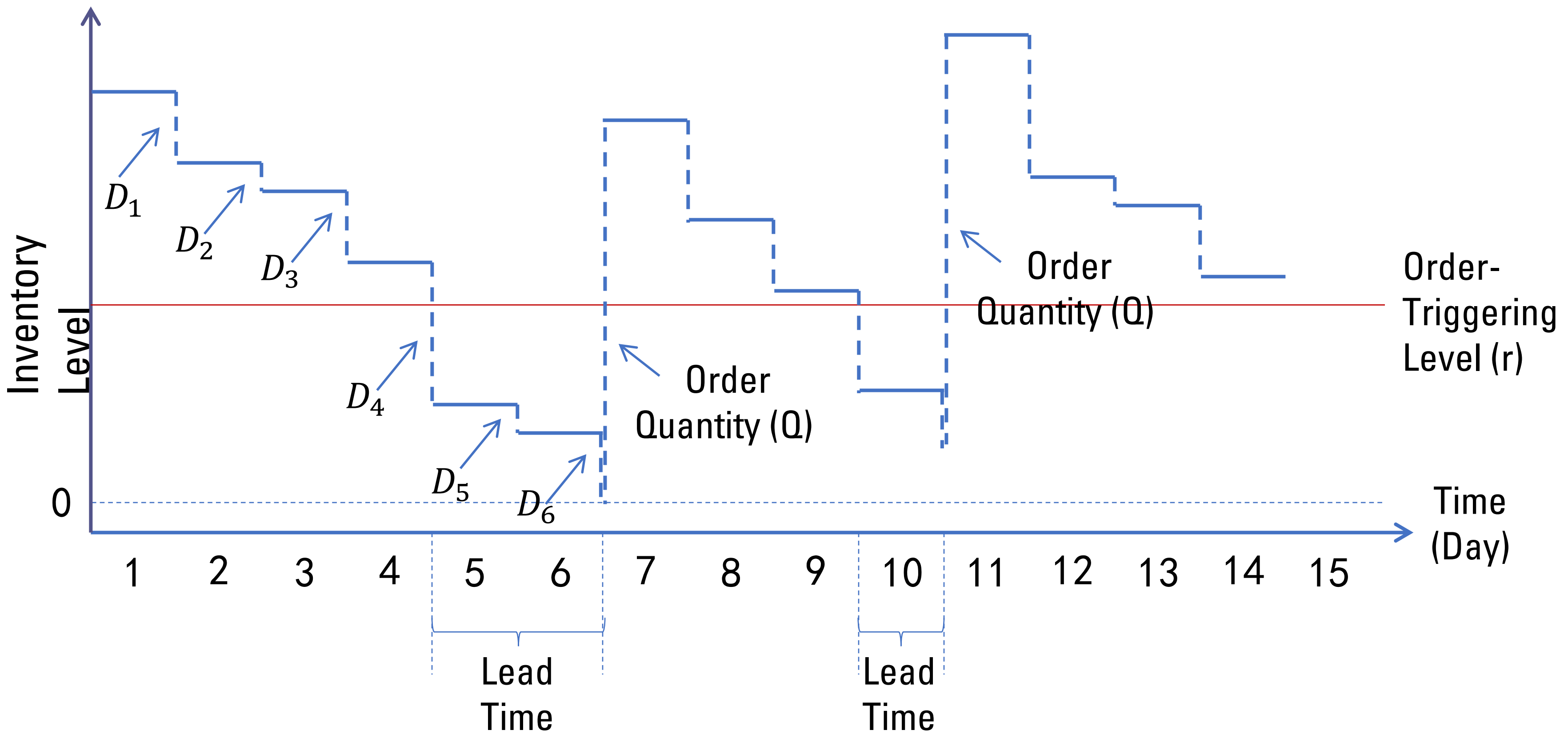


Inventory Management

- The goal of inventory management is to achieve an optimal balance between inventory holding cost, ordering cost, and stock-out cost.
- The biggest challenges are posed by uncertainties
  - Demand uncertainty
  - Lead time uncertainty
- The early decisions can affect the future decisions.
- There are two types of commonly used policies:
  - For continuous inventory monitoring, set up an order-triggering level (r) and decide the order quantity (Q)
  - For periodic inventory monitoring, set up an order-triggering level (s) and an order-up-to level (S)

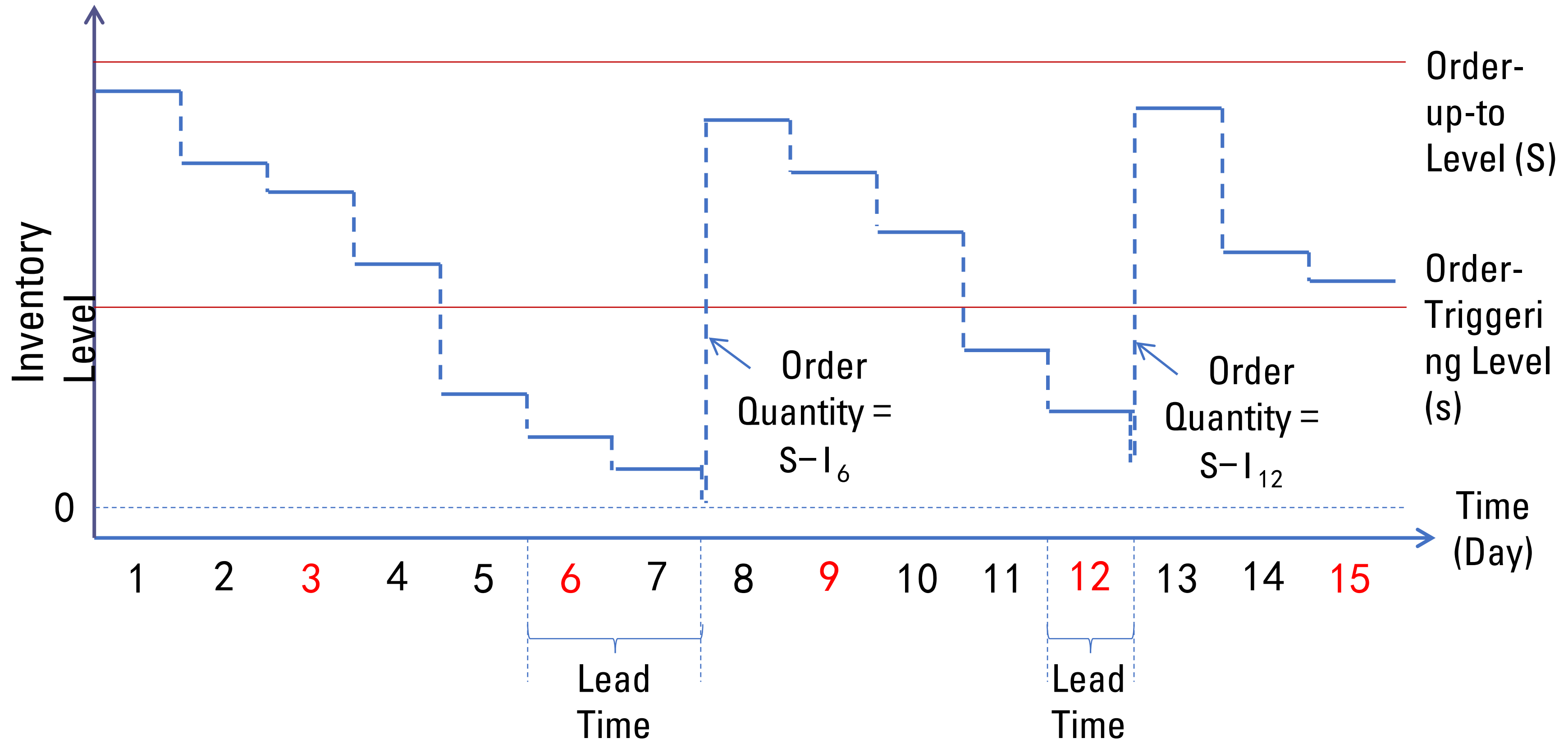
Inventory Management

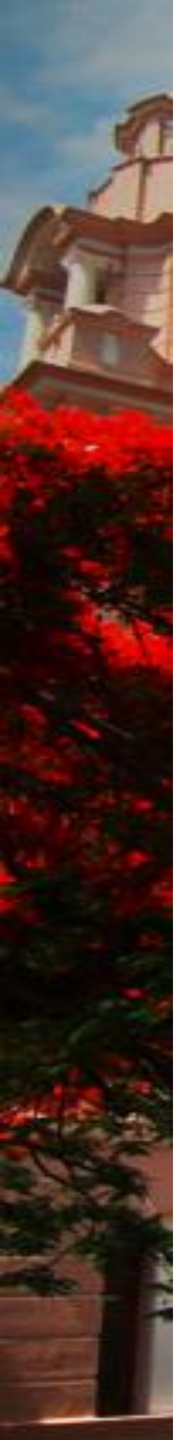
- Continuous inventory monitoring: (r,Q) policy



Inventory Management

- Periodic inventory monitoring: (s,S) policy



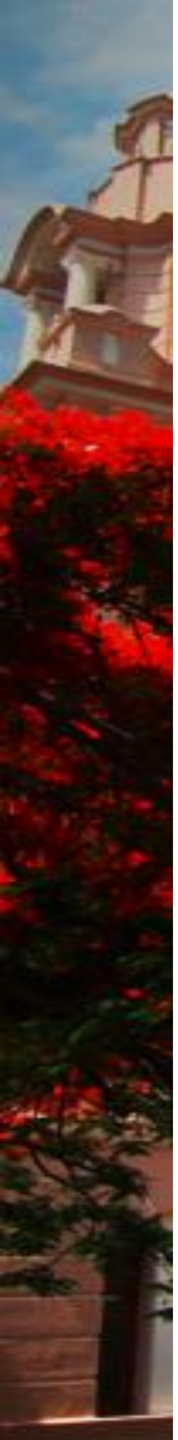


Revenue Management

- The key decisions for revenue management include the pricing and/or the allocation of a limited resource (e.g., airline tickets and hotel rooms).
- The key to the optimal decisions is quantifying the opportunity cost of selling the resource. However, when demand is stochastic, how to determine the opportunity cost of selling the resource?
- The key question is: the opportunity cost of selling depends on your policy!
  - Dynamic production capacity allocation: Customer orders (with differing prices and quantities) arrive randomly over time. Your loss of selling the production capacity to the current customer depends on to which future customers you allocate the capacity.
  - Dynamic fashion apparel pricing: Customers with different WTP arrive randomly over time. Your loss of selling a unit at the current price depends on your future pricing decisions.

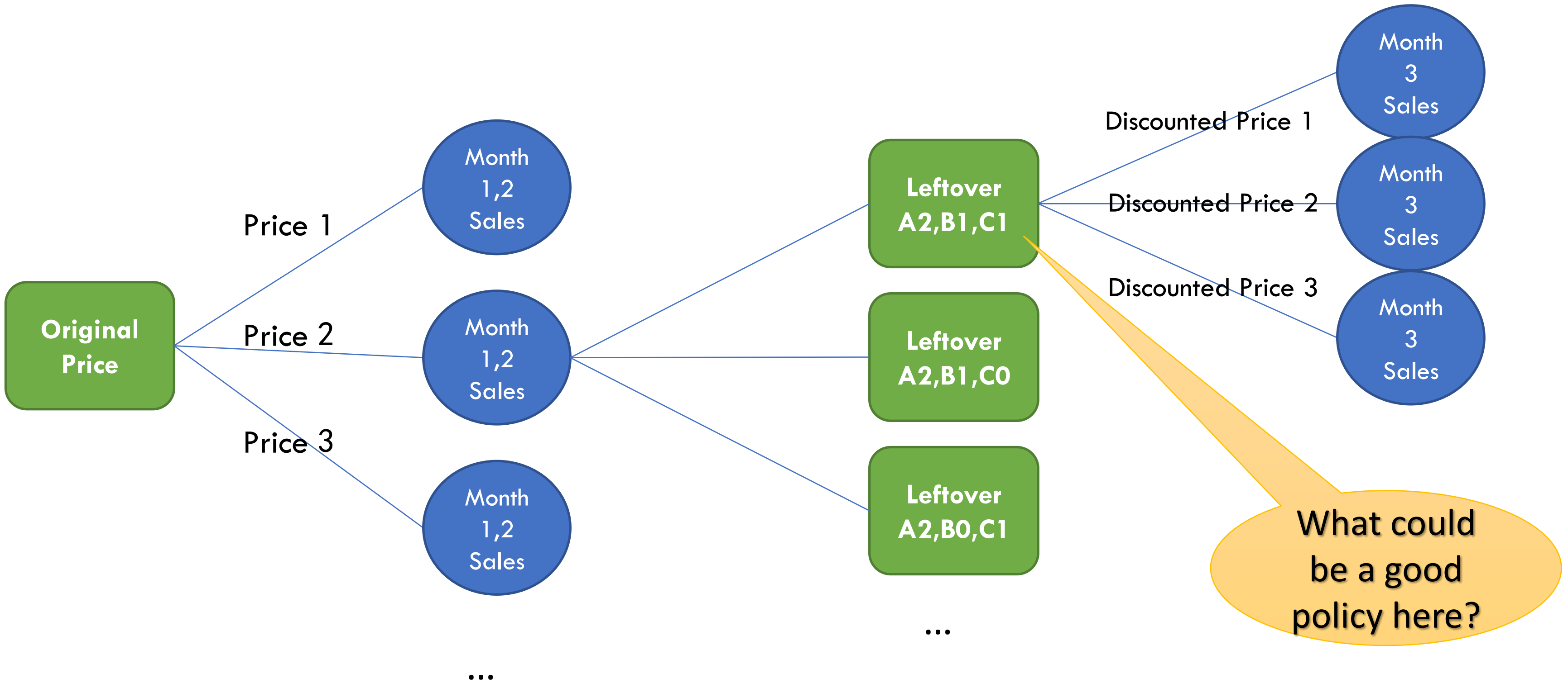
K-Fashion Apparel Pricing

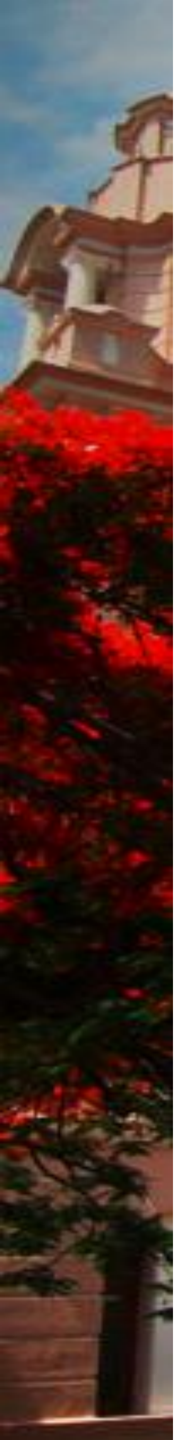
- K-Fashion is a boutique store for women's fashion apparel located in a big shopping mall. The store is targeting young female white-collar who care less about brand but more about fashion and price.
- For the next season (3 month), K-Fashion has ordered many different stock keeping units (SKUs) from a foreign supplier. Due to the long production and order lead time, K-Fashion can place the order only once. Given the large store traffic, the store ordered 10 pieces for each SKU.
- There are three SKUs—A, B, and C—of a particular style. The sale of this style is independent of other styles. Assume that the demand for each SKU is independent. For example, customers that suit size L will never buy size M. In other words, if a customer who intends to buy size L finds that size L is not available, the s/he will walk away without buying other sizes.
- Customers arrive randomly. Historical data suggests that the traffic is smaller in the first two months and larger in the last one. A customer's willingness to pay is also random. It is likely to be higher in the first two month and lower in the last month.
- The manager's job is to determine the tag price of this style for the first two months and a discounted price for the last month. Any unsold inventory after the season will be discarded with zero value. The constraint is that you must set the same price for all the three SKUs as they differ only in color or size. The price must end with 9.



K-Fashion Apparel Pricing

- Decision tree (partial illustration):





In-Class Exercise: Café du Donut

- The Café buys donuts each day for $40 per carton of 20 dozen donuts. Any cartons not sold are thrown away at the end of the day. If a carton is sold, the total revenue is $60. How to optimize the daily order quantity? Suppose only 5, 6, and 7 cartons are allowed.

DAILY DEMAND (CARTONS)		PROBABILITY	CUMULATIVE PROBABILITY
4		0.05	0.05
5		0.15	0.2
6		0.15	0.35
7		0.20	0.55
8		0.25	0.8
9		0.10	0.9
10		0.10	1.0
	Total	1.00	

What is the EVPI?

In-Class Exercise: Café du Donut

- Assume demand is not independent. If today's demand is X, then tomorrow's demand will be randomly drawn again with a 50% chance and will be 14 – X with a 50% chance.
- If a customer finds that donuts are sold out, s/he will buy from a nearby bakery shop.
- Assume you know yesterday's demand is 5. How to optimize today's order quantity?