Lecture 11 Integer Optimization

Yingda Zhai (Dr.)

School of Computing, NUS

November 1, 2022

BT1101 Roadmap: Predictive (7-10), Prescriptive (11-12)

Week 1 - 6	Descriptive Analytics
Week 7	Linear Regression
Week 8	Logistic Reg & Time Series
Week 9	Data Mining Basics
Week 10	*Assessment*
Week 11	Linear Optimization
Week 12	Integer Optimization & Summary
Week 13	Tutorials and Consultation
Nov 24	*Final Exam*

Learning Objectives

- Understand when and how integer constraints apply to linear optimization problem.
- Be able to know how to use linear-program relaxation (LPR) to gain insight into integer-constrainted problems.
- Be able to solve integer-constrained optimization problems in R, including logical constraints with binary decisions.

Integer Optimization

Roadmap

Identify	Identify decision variables, objective function and set of constraints.
Formulate	Formulate the integer optimization problem.
Solve	Solve it by graph or (mainly) in R.
Compare	Conduct sensitivity analysis.
Interpret	Writing recommendation and interpretation.

Decision Variables that are Integer-valued

- In linear optimization problems we covered last week, all decision variables are assumed to be *continuous*, or real-valued numbers, i.e., $x \in \mathbb{R}$.
- For instance, recall in Vegan's problem, the optimal diet plan was: $x_1 = 5.93$ unit of soybeans; $x_2 = 3.70$ unit of parsnip and $x_3 = 0$ unit of kale.
- What if the diet plan requires integer-valued consumption?
- In practice, many optimization problems require the decision variables to be integer-valued, i.e. $x \in \mathbb{N}$, e.g.
  - the number of machines running;
  - the number of cars produced;
  - the number of shifts assigned to an employee, etc.
- In an optimization problem where some of the decision variables must be integer-valued, it is called a mixed-integer optimization problem.

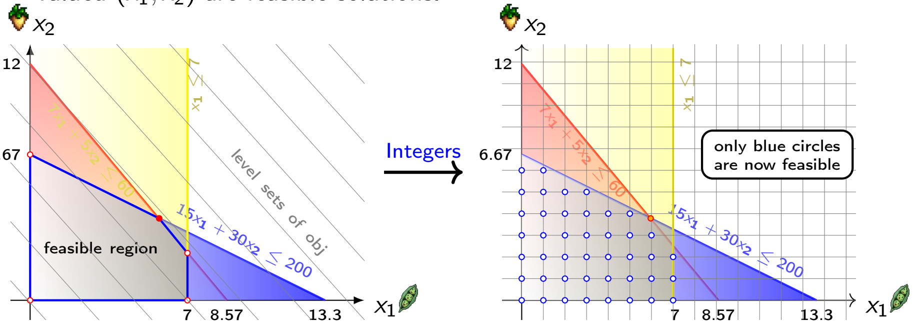
Formulation of Vegan's Utility Maximization Problem

- Recall in Vegan's problem, our decision variables are:
  - x_1: piece of soybean consumed (integer).
  - x_2: piece of parsnip consumed (integer).
  - x_3: piece of kale consumed (integer).
- With integer-valued decision variables, one additional constraints is needed for linear problem formulation:

Linear Problem	Math Expression
Objective function:	
Maximize utility choosing $\boldsymbol{x}$	$F(\boldsymbol{x};\theta) = 250x_1 + 225x_2 + 300x_3$
Set of constraints:	*subject to*
Budget constraint	$7x_1 + 5x_2 + 8x_3 \leq 60$
Storage constraint	$15x_1 + 30x_2 + 40x_3 \leq 200$
Diet constraint	$x_1 \leq 7$
Non-negativity	$x_1 \geq 0$, $x_2 \geq 0$, and $x_3 \geq 0$
Integer constraint	x_1, x_2 and x_3 are integer.

Vegan's Problem: Integer Constraint in Graph

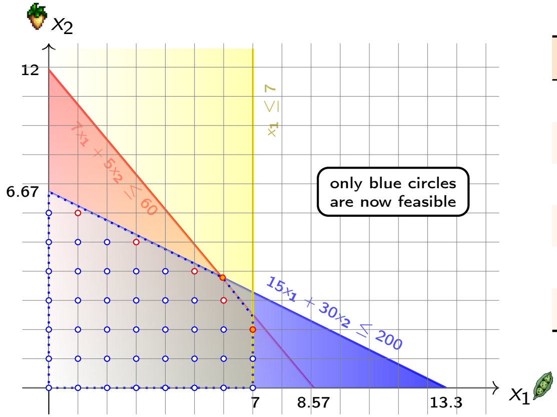
- Recall we plotted constraints onto (x_1, x_2)-coordinates. Now only integer valued (x_1, x_2) are feasible solutions.



- After integer constraint being imposed, only integer-valued solutions ○ that lie within the "feasible region" are feasible.
- Feasible set now consists of all those integer consumptions ○.
- The once optimal real-valued solution ● is even not feasible.

Vegan's Problem: Integer Constraint in Graph

- The real-valued optimal solution is $x_1 = 5.93$, $x_2 = 3.70$ and $x_3 = 0$.
- Could we simply round off the real-valued solution?
  No. In this case $(6, 4)$ is not even feasible.
- Could we round down or round off to the nearest feasible solution?
  Maybe. The nearest feasible solutions are not guaranteed to be optimal!



Feasible Soln	Utility (obj)
● (5.93, 3.70)	2315
○ (1, 6)	1600
○ (3, 5)	1875
○ (5, 4)	2150
○ (6, 3)	2175
● (7, 2)	\$2200

RM: Note we keep $x_3 = 0$.

Linear Program Relaxation and Insight for Integer Optimization

- Solving an integer optimization problem **without** the integer constraint is called linear program relaxation (back to week 11) since we temporarily "relax" the optimization problem from the integer constraint.
  - If the solution to LP relaxation happens to be integer-valued (lucky hit!), such solution must be optimal to the integer optimization problem. e.g. Farmer Jean's problem ($x_1 = 80$, $x_2 = 120$) .
  - If not, the real-valued solution is used as a starting point to search for integer solution. e.g. "branch-and-bound" algorithm used in `lpSolve`.
- What's the insights from LP relaxation for integer problem?
  1. The optimal value of objective function of the LP relaxation must be at least large as that of corresponding integer-constrained problem.
  2. Rounding off might not be feasible, nor guarantee to be optimal.
  3. The integer constraint might change which decision variables are relevant at optimum!
  - It turns out that the optimal solution for Vegan's integer optimization problem is $x_1 = 6$, $x_2 = 2$ and $x_3 = 1$!
    (far from $x_1 = 5.93$, $x_2 = 3.70$, $x_3 = 0$ before)

Coding Integer Constraints into `lpSolve` in R

Maximize utility choosing $\boldsymbol{x}$	$F(\boldsymbol{x};\theta) = 250x_1 + 225x_2 + 300x_3$
Budget constraint	$7x_1 + 5x_2 + 8x_3 \leq 60$
Storage constraint	$15x_1 + 30x_2 + 40x_3 \leq 200$
Diet constraint	$x_1 + \qquad + \qquad \leq 7$
Non-negativity	$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0$
Integer Constraints	x_1, x_2 and x_3 are integers.

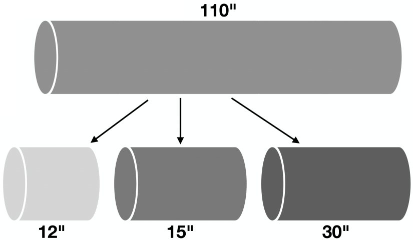
```r
# first define all parameters
objective.fn = c(250, 225, 300)
const.mat = matrix(c(7, 5, 8, 15, 30, 40, 1, 0, 0),
                   ncol = 3, byrow = TRUE)
const.dir = c("<=", "<=", "<=")
const.rhs = c(60, 200, 7)
# then solve model
lp.solution = lp("max", objective.fn, const.mat,
                 const.dir, const.rhs,
                int.vec = c(1,2,3), compute.sens = FALSE)
```

RM: Sensitivity analysis is invalid in integer optimization.

Two Examples in Integer Optimization

Factory Metal Cutting Problem

- A factory specializes in cutting 110 inch rolls of sheet metal into smaller rolls of 12, 15 and 30 rolls.
- There are several different cutting patterns (leaving different sizes of scraps), which is listed in the table below, along with demand information next week.



Pattern	12"	15"	30"	Scrap
P1	0	7	0	5
P2	0	1	3	5
P3	1	0	3	8
P4	9	0	0	2
P5	2	1	2	11
P6	7	1	0	11
Demand	500	715	630	-

- How many 110" rolls should we cut into each of the six patterns to meet the demand with minimized scraps? (Example 12.5 from JE)

Factory Metal Cutting Problem Formulation

- How many 110" rolls should we cut into each of the six patterns to meet the demand with minimized scraps?
- Decision Variables: x_i, the number of 110" rolls cut into pattern i, $i = 1, 2, \dots, 6$.

Pattern	12"	15"	30"	Scrap
P1	0	7	0	5
P2	0	1	3	5
P3	1	0	3	8
P4	9	0	0	2
P5	2	1	2	11
P6	7	1	0	11
Demand	500	715	630	-

Minimize total scrap with $\boldsymbol{x}$	$F(\boldsymbol{x};\theta) = 5x_1 + 5x_2 + 8x_3 + 2x_4 + 11x_5 + 11x_6$
Demand Constraint for 12"	$0x_1 + 0x_2 + 1x_3 + 9x_4 + 2x_5 + 7x_6 \geq 500$
Demand Constraint for 15"	$7x_1 + 1x_2 + 0x_3 + 0x_4 + 1x_5 + 1x_6 \geq 715$
Demand Constraint for 30"	$0x_1 + 3x_1 + 3x_3 + 0x_4 + 2x_5 + 0x_6 \geq 630$
Non-negativity	$x_i \geq 0$, for $i = 1, \dots, 6$
Integer Constraints	x_i for all i are integers.

Factory Metal Cutting Problem Coded in R

Minimize total scrap with $\boldsymbol{x}$	$F(\boldsymbol{x};\theta) = 5x_1 + 5x_2 + 8x_3 + 2x_4 + 11x_5 + 11x_6$
Demand Constraint for 12"	$0x_1 + 0x_2 + 1x_3 + 9x_4 + 2x_5 + 7x_6 \geq 500$
Demand Constraint for 15"	$7x_1 + 1x_2 + 0x_3 + 0x_4 + 1x_5 + 1x_6 \geq 715$
Demand Constraint for 30"	$0x_1 + 3x_1 + 3x_3 + 0x_4 + 2x_5 + 0x_6 \geq 630$
Non-negativity	$x_i \geq 0$, for $i = 1, \ldots, 6$
Integer Constraints	x_i for all i are integers.

```
objective.fn = c(5, 5, 8, 2, 11, 11)
const.mat = matrix(c(0, 0, 1, 9, 2, 7,
                     7, 1, 0, 0, 1, 1,
                     0, 3, 3, 0, 2, 0) , ncol=6 , byrow=TRUE)
const.dir = c(">=", ">=", ">=")
const.rhs = c(500, 715, 630)
# solve linear program with integer constraint
lp.solution = lp("min", objective.fn, const.mat,
                  const.dir, const.rhs, int.vec = c(1,2,3,4,5,6))
# optimal solution
print(lp.solution$solution)
>[1]   73 210 0 56 0 0
```

RM: "Meet the demand" means that produced rolls have to at least many as required (`">="`).

Binary Decision Variables and Logical Constraints

- A special class of integer-valued decision variables are binary decision variables, i.e. $x \in \{0, 1\}$. e.g. pass/fail, invest/not, like/unlike, etc.
- Integer optimization is more capable than you might expect:
- Logical constraints can be formalized as linear constraints with **binary** decisions. For instance, "if x_1 is chosen, x_2 must be then chosen as well". This logical statement can be written as the following constraint:

$$x_2 - x_1 \geq 0 \qquad \text{(if } x_1 = 1,\ x_2 \text{ has to be 1 by constraints)}$$

- The following table (*12.5 from JE*) shows some examples:

Logical Statement	Alternative	Integer Constraint
If A, then B	$B \geq A$	$B - A \geq 0$
If not A, then B	$B \geq (1 - A)$	$A + B \geq 1$
If A, then not B	$(1 - B) \geq A$	$A + B \leq 1$
At most one A or B	$A + B \leq 1$	$A + B \leq 1$
If A, then B and C	$B \geq A$ & $C \geq A$	$B + C - 2A \geq 0$
If A and B, then C	$C \geq (A + B - 1)$	$A + B - C \leq 1$

RM: A, B and C are **binary** decisions, taking value either 0 or 1.

Example B: Where To Locate Distribution Centers

- Let's see how we can use binary decision variables in optimization problem.
- Companies such as Walmart, Amazon, need to choose their distribution centers to minimize shipping cost. There are four (indexed by i) cities to build a distribution center which serves five (indexed by j) cities. Each city needs to be served by at least one distribution center while one distribution can serve multiple cities.
- The following table summarizes the cost c_{ij} to send goods from distribution center i to city j.

Location	Houston ($j = 1$)	Les Vegas ($j = 2$)	New Orleans ($j = 3$)	Chicago ($j = 4$)	Portland ($j = 5$)
Irvine ($i = 1$)	\$40,000	\$11,000	\$75,000	\$70,000	\$60,000
Denver ($i = 2$)	\$72,000	\$77,000	\$120,000	\$30,000	\$75,000
Deland ($i = 3$)	\$24,000	\$44,000	\$45,000	\$80,000	\$90,000
Austin ($i = 4$)	\$32,000	\$55,000	\$90,000	\$20,000	\$105,000

- We only have enough capital to build at most two distribution centers, which cities should we choose to minimize the total shipping cost?

Distribution Center Problem Formulation

- The tricky part of this problem is that we not only decide which cities to build distribution centers but need to assign cities to which distribution center they receive their goods from.
- Decision variables:
  - $x_{ij} = 1$ if distribution center i is assigned to serve city j;
  - $y_i = 1$ if location i is chosen to build a distribution center.

Minimize total shipping cost, choosing $(\boldsymbol{x}, \boldsymbol{y})$	$\min_{(\boldsymbol{x},\boldsymbol{y})} \sum_i \sum_j c_{ij} x_{ij}$	
Set of constraint	*subject to*	(#const)
"each city served by at least one center"	$\sum_i x_{ij} \geq 1,\ \forall j{=}1,\ldots,5$	5
"budget for at most two distribution centers"	$\sum_i y_i \leq 2$	1
"an *open* center for its serving cities", i.e. if $x_{ij} = 1$, then y_i needs to be 1.	$y_i - x_{ij} \geq 0$, $\forall i{=}1,\ldots,4,\ \forall j{=}1,\ldots,5$	20
binary decision variables	$x_{ij}, y_i \in \{0, 1\},\ \forall i,j$	24

- It's essential to make set of constraints right!

RM: There are total 26 constraints (row) and 24 decision variables (col) in `const.mat`. Watch out!

Distribution Center Problem Coded into R

- It can be tricky to code the problem into R, especially the `const.mat`.
  - **All** decision variables (in total 24), (x_{ij}, y_i), need to show up in a consistent order in the constraint matrix. One variable, one column.
  - One row of the constraint matrix represents one constraint. There are 26 of them except for binary decision variable constraints.
  - Use `binary.vec` to impose the binary decision variable constraints. In this case, all decisions are binary, `binary.vec = c(1:24)`.
  - Write the constraint down in paper to make sure everything remains crystal clear.
- "*each city j needs to be served by at least one center*" (first 5 constraints), $\sum_i x_{ij} \geq 1$. Constraint for $j = 1$ is shown below; repeat for $j = 2, 3, 4, 5$.

$$
\begin{array}{l}
1\,x_{11} + 0\,x_{12} + 0\,x_{13} + 0\,x_{14} + 0\,x_{15} + \\
1\,x_{21} + 0\,x_{22} + 0\,x_{23} + 0\,x_{24} + 0\,x_{25} + \\
1\,x_{31} + 0\,x_{32} + 0\,x_{33} + 0\,x_{34} + 0\,x_{35} + \\
1\,x_{41} + 0\,x_{42} + 0\,x_{43} + 0\,x_{44} + 0\,x_{45} + \\
0\;y_1 + 0\;y_2 + 0\;y_3 + 0\;y_4 \geq 1
\end{array}
$$

$$
\begin{pmatrix}
1, & 0, & 0, & 0, & 0, \\
1, & 0, & 0, & 0, & 0, \\
1, & 0, & 0, & 0, & 0, \\
1, & 0, & 0, & 0, & 0, \\
0, & 0, & 0, & 0 &
\end{pmatrix}
$$

RM: Each single decision variable needs to be there even though they are zeros.

RM: This is actually one long **row vector** rather than a matrix.

Distribution Center Problem Coded into R

- It can be tricky to code the problem into R, especially the `const.mat`.
  - **All** decision variables (in total 24), (x_{ij}, y_i), need to show up in a consistent order in the constraint matrix. One variable, one column.
  - One row of the constraint matrix represents one constraint. There are 26 of them except for binary decision variable constraints.
  - Use `binary.vec` to impose the binary decision variable constraints. In this case, all decisions are binary, `binary.vec = c(1:24)`.
  - Write the constraint down in paper to make sure everything remains crystal clear.
- "*budget for at most two distribution centers*" (the constraint in the middle), $\sum_i y_i \leq 2$.

$$\begin{array}{l}
0\, x_{11} + 0\, x_{12} + 0\, x_{13} + 0\, x_{14} + 0\, x_{15} + \\
0\, x_{21} + 0\, x_{22} + 0\, x_{23} + 0\, x_{24} + 0\, x_{25} + \\
0\, x_{31} + 0\, x_{32} + 0\, x_{33} + 0\, x_{34} + 0\, x_{35} + \\
0\, x_{41} + 0\, x_{42} + 0\, x_{43} + 0\, x_{44} + 0\, x_{45} + \\
1\, y_1 + 1\, y_2 + 1\, y_3 + 1\, y_4 \leq 2
\end{array}$$

$$\begin{pmatrix}
0, & 0, & 0, & 0, & 0, \\
0, & 0, & 0, & 0, & 0, \\
0, & 0, & 0, & 0, & 0, \\
0, & 0, & 0, & 0, & 0, \\
1, & 1, & 1, & 1 &
\end{pmatrix}$$

RM: Each single decision variable needs to be there even though they are zeros.

RM: This is actually one long **row vector** rather than a matrix.

Distribution Center Problem Coded into R

- It can be tricky to code the problem into R, especially the `const.mat`.
  - **All** decision variables (in total 24), (x_{ij}, y_i), need to show up in a consistent order in the constraint matrix. One variable, one column.
  - One row of the constraint matrix represents one constraint. There are 26 of them except for binary decision variable constraints.
  - Use `binary.vec` to impose the binary decision variable constraints. In this case, all decisions are binary, `binary.vec = c(1:24)`.
  - Write the constraint down in paper to make sure everything remains crystal clear.
- "*an open center for its serving cities*" (last 20 constraints), $y_i - x_{ij} \geq 0$, for $i = 1, \ldots, 4$ and $j = 1, \ldots, 5$. $i = 1, j = 1$ is shown below, enumerate others.

$$\begin{array}{cccccccccc}
-1\,x_{11} & + & 0\,x_{12} & + & 0\,x_{13} & + & 0\,x_{14} & + & 0\,x_{15} & + \\
0\,x_{21} & + & 0\,x_{22} & + & 0\,x_{23} & + & 0\,x_{24} & + & 0\,x_{25} & + \\
0\,x_{31} & + & 0\,x_{32} & + & 0\,x_{33} & + & 0\,x_{34} & + & 0\,x_{35} & + \\
0\,x_{41} & + & 0\,x_{42} & + & 0\,x_{43} & + & 0\,x_{44} & + & 0\,x_{45} & + \\
1\;y_1 & + & 0\;y_2 & + & 0\;y_3 & + & 0\;y_4 & \geq & 0 &
\end{array}$$

RM: Each single decision variable needs to be there even though they are zeros.

$$\begin{pmatrix}
-1, & 0, & 0, & 0, & 0, \\
0, & 0, & 0, & 0, & 0, \\
0, & 0, & 0, & 0, & 0, \\
0, & 0, & 0, & 0, & 0, \\
1, & 0, & 0, & 0 &
\end{pmatrix}$$

RM: This is actually one long **row vector** rather than a matrix.

Distribution Center Problem Coded into R

```
const.mat = matrix(c( ## 26 by 24 matrix, const by variable ##
  # constraints 1-5: "each city served by one center"
  rep(c(1,0,0,0,0), 4), rep(0,4),
  rep(c(0,1,0,0,0), 4), rep(0,4),
  rep(c(0,0,1,0,0), 4), rep(0,4),
  rep(c(0,0,0,1,0), 4), rep(0,4),
  rep(c(0,0,0,0,1), 4), rep(0,4),
  # constraint 6: "budget for at most 2 centers"
  rep(c(0,0,0,0,0), 4), rep(1,4),
  # constraints 7-26: "an open center for served cities"
  rep(0,0), -1, rep(0,19), rep(0, 0), 1, rep(0, 3),
  rep(0,1), -1, rep(0,18), rep(0, 0), 1, rep(0, 3),
  rep(0,2), -1, rep(0,17), rep(0, 0), 1, rep(0, 3),
  rep(0,3), -1, rep(0,16), rep(0, 0), 1, rep(0, 3),
  rep(0,4), -1, rep(0,15), rep(0, 0), 1, rep(0, 3),
  rep(0,5), -1, rep(0,14), rep(0, 1), 1, rep(0, 2),
  rep(0,6), -1, rep(0,13), rep(0, 1), 1, rep(0, 2),
  rep(0,7), -1, rep(0,12), rep(0, 1), 1, rep(0, 2),
  rep(0,8), -1, rep(0,11), rep(0, 1), 1, rep(0, 2),
  rep(0,9), -1, rep(0,10), rep(0, 1), 1, rep(0, 2),
  rep(0,10), -1, rep(0,9), rep(0, 2), 1, rep(0, 1),
  rep(0,11), -1, rep(0,8), rep(0, 2), 1, rep(0, 1),
  rep(0,12), -1, rep(0,7), rep(0, 2), 1, rep(0, 1),
  rep(0,13), -1, rep(0,6), rep(0, 2), 1, rep(0, 1),
  rep(0,14), -1, rep(0,5), rep(0, 2), 1, rep(0, 1),
  rep(0,15), -1, rep(0,4), rep(0, 3), 1, rep(0, 0),
  rep(0,16), -1, rep(0,3), rep(0, 3), 1, rep(0, 0),
  rep(0,17), -1, rep(0,2), rep(0, 3), 1, rep(0, 0),
  rep(0,18), -1, rep(0,1), rep(0, 3), 1, rep(0, 0),
  rep(0,19), -1, rep(0,0), rep(0, 3), 1, rep(0, 0)),
  ncol=24 , byrow=TRUE)
```

Optimal Location for Distribution Centers

```
objective.fn = c(40, 11, 75, 70, 60,
                 72, 77, 120, 30, 75,
                 24, 44, 45, 80, 90,
                 32, 55, 90, 20, 105,
                 0, 0, 0, 0)
const.mat = matrix(c(...) , ncol=24 , byrow=TRUE)
const.dir = c(rep(">=",5), "<=", rep(">=", 20))
const.rhs = c(rep(1,5), 2, rep(0,20))
# solve model
lp.solution = lp("min", objective.fn, const.mat,
                 const.dir, const.rhs, binary.vec = c(1:24))
# display optimal decision for where to build distribution centers
lp.solution$solution[21:24]
> [1] 1 0 0 1
```

- `solution[21:24]` contains the optimal solution for location of distribution centers, i.e. Irvine and Austin.

Location	Houston	Les Vegas	New Orleans	Chicago	Portland
Irvine ($i = 1$)	\$40,000	\$11,000	\$75,000	\$70,000	\$60,000
Denver ($i = 2$)	\$72,000	\$77,000	\$120,000	\$30,000	\$75,000
Deland ($i = 3$)	\$24,000	\$44,000	\$45,000	\$80,000	\$90,000
Austin ($i = 4$)	\$32,000	\$55,000	\$90,000	\$20,000	\$105,000

Optimal Assignment of Cities for Distribution Centers

```
# optimal assignment for Irvine
lp.solution$solution[1:5]
> [1] 0 1 1 0 1
```

```
# optimal assignment for Austin
lp.solution$solution[16:20]
> [1] 1 0 0 1 0
```

- `solution[1:5]` : Las Vegas, New Orleans and Porland should be served by Irvine.
- `solution[16:20]` : Houston and Chicago should be served by Austin.

Location	Houston	Les Vegas	New Orleans	Chicago	Portland
Irvine ($i = 1$)	\$40,000	\$11,000	\$75,000	\$70,000	\$60,000
Denver ($i = 2$)	\$72,000	\$77,000	\$120,000	\$30,000	\$75,000
Deland ($i = 3$)	\$24,000	\$44,000	\$45,000	\$80,000	\$90,000
Austin ($i = 4$)	\$32,000	\$55,000	\$90,000	\$20,000	\$105,000

```
# display the minimized total shipping cost at the optimal solution
lp.solution
> Success:  the objective function is 198
```

- The total shipping cost (minimized) at the optimal solution is \$198k.

Summary

- In today's lecture we discussed how to extend linear program (LP) to solve integer-valued optimization problems where some or all decision variables have to be integers, or even binary variables (yes/no decisions).
- Logical constraint can be written as linear constraints with binary decision variables. Many real world problems can be formalized as linear programs, given LP's yet simple forms.
- We've seen working example with 20+ decision variables. Keep in mind that in reality, the solution dimension of linear optimization problems can be blown up very quickly!
- The way we set up LP, the solution, and the sensitivity analysis could be very useful in fact-based decision making process, which is exactly our goal behind *prescriptive analytics*.

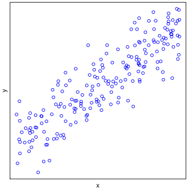
Predictive and Prescriptive Analytics Recap

Linear Regression and Logistic Regression

1. Students should be able to understand and use both simple and multivariate linear regressions to estimate trends.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \epsilon$$

Interpretation: ◦ continuous IVs ◦ categorical (binary) IVs ◦ nonlinear and interaction terms



Assumptions!

```
Call:
lm(formula = wage ~ hours + educ + hours:educ, data = mroz)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.5879198  0.7321832  -3.535 0.000434 ***
hours        0.0001089  0.0006804   0.160 0.872936
educ         0.3142493  0.0595966   5.273 1.76e-07 ***
hours:educ   0.0001096  0.0000542   2.023 0.043418 *
```

2. Be able to interpret the output of a regression model, including regression coefficients, p-values, confidence intervals, hypothesis testing about coefficients, and goodness-of-fit statistics.

$$\text{logit}(p) = \log \frac{p}{1-p} = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k$$

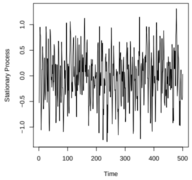
3. Understand and be able to use logistic regression to estimate categorical dependent variables (i.e. classification).

Time Series Forecasting

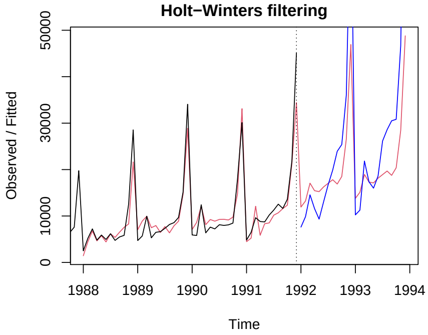
1. Understand basic concepts that are important to time-series analysis such as difference between time series and cross-sectional data, stationarity, weakly dependence, trend.

$$Y_t = \beta_0 + \beta_1 X_{1t} + \cdots + \beta_k X_{kt} + \gamma t + \epsilon_t$$

Understand: ◦ stationary ◦ weakly dependent ◦ trend and de-trend ◦ spurious regression



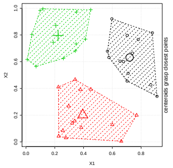
white-noise w_t



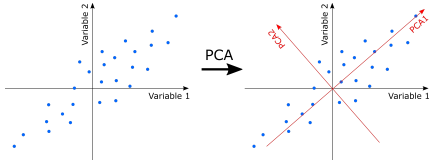
2. Understand the concept for a family of smoothing models, e.g., moving-averages, Holt-Winters.
3. Be able to apply the suitable smoothing model for trending/seasonal time-series for forecasting.

Week 9 - Model Selection, Data Reduction and Classification

1. Understand the concept and best practice of model selection; Be able to conduct F-test on linear combination of regression coefficients and forward/backward stepwise model selection.
   - Restricted vs unrestricted
   - Chow test
   - backward/forward stepwise selection



k-mean clustering



2. Be able to apply principal component analysis to summarize information in a large dataset. Understand the similarity and difference between model selection and dimensionality reduction.

	Actual: Yes	Actual: No
Pred: Yes	true positive (TP)	false positive (FP)
Pred: No	false negative (FN)	true negative (TN)

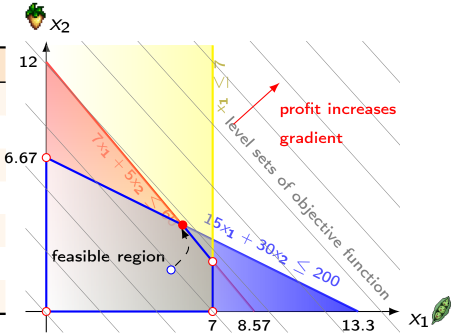
- sensitivity
- precision
- accuracy
- type-I and type-II errors

3. Understand clustering such as *k*-mean and logistic regression as classification; Evaluate classification via confusion matrix.

Week 11 - Linear Optimization and Sensitivity Analysis

1. Be able to identify the objective function, the set of constraints and formulate linear optimization problem.
   - feasible region
   - binding constraint
   - level set and gradient
   - corner solution

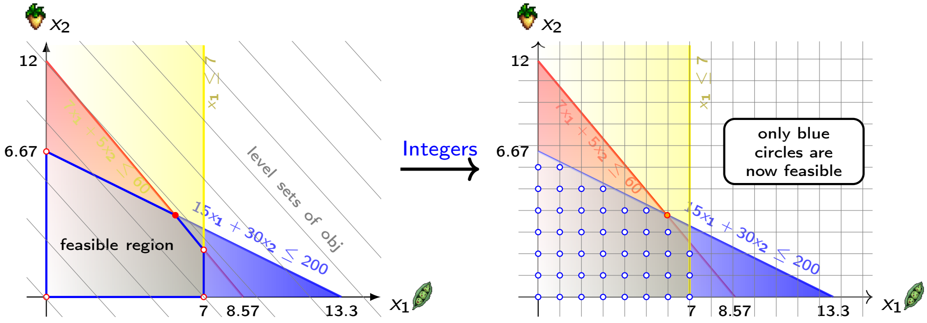
Linear Problem	Math Expression
Objective function:	
Maximize utility	$F = 250x_1 + 225x_2 + 300x_3$
Set of constraints:	
Budget constraint	$7x_1 + 5x_2 + 8x_3 \le 60$
Storage constraint	$15x_1 + 30x_2 + 40x_3 \le 200$
Diet constraint	$x_1 \le 7$
Non-negativity	$x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0$



2. Be able to solve linear optimization problem with intuition and R.
3. Be comfortable in interpreting the result of optimization and of sensitivity analysis and writing "prescriptive" recommendation with justification.
   - changes in objective function coefficients
   - shadow price of constraint

Week 12 - Integer Optimization and Binary Decisions

1. Understand when and how integer constraints apply to linear optimization problem.
2. Be able to know how to use linear-program relaxation (LPR) to gain sight into integer-constrainted problems.

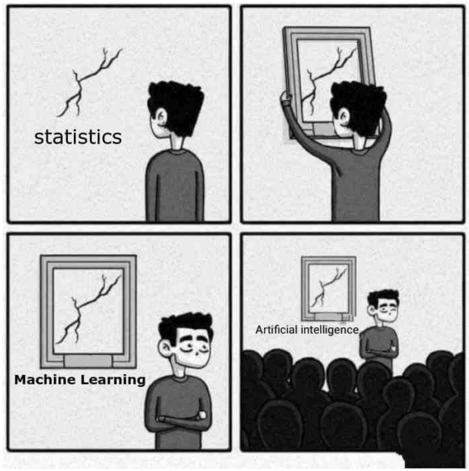


3. Be able to solve integer-constrained optimization problems, especially with binary decisions and logical constraints.

Course Summary

1. Predictive analytics builds various models to understand relationships observed in the data, whether using linear regression, time-series, clustering and PCA.
2. Prescriptive analytics formulates decision-making process in a business context as a linear optimization problem with objective and a set of constraints and allows us to obtain optimal decisions, responding to changing enviornment.

- In your future courses and career, you'll build upon these techniques: learning non-parametric and Bayesian statistics, powerful simulation and optmization toolkits, various machine learning techniques.
- Don't be overwhelmed by techniques but learn their nature (concept) and use them as a way to understand our data and theories.


statistics
Machine Learning
Artificial intelligence

Course Logistics

BT1101 Roadmap: Predictive (7-10), Prescriptive (11-12)

Week 1 - 6	Descriptive Analytics
Week 7	Linear Regression
Week 8	Logistic Reg & Time Series
Week 9	Data Mining Basics
Week 10	*Assessment*
Week 11	Linear Optimization
Week 12	Integer Optimization & Summary
Week 13	Tutorials and Consultation
Nov 24	Final Exam

Final Exam (40%)

- Time: **Nov 24**, **9:00 - 11:30 AM**.
- Where: Examplify (ExamSoft) in classroom setup.
- Tentative Format:
  1. 20 Multiple Choice (30 min);
  2. 3 Comprehensive Questions (120 min): both R coding and short answer questions.

- We shall post a practice exam.
- Logistic details will be posted later in Canvas.

Additional Consultation Hours

- All consultation sessions before the final:
  - Session A: **Nov 14 Mon**, 3:00 - 5:00 PM
  - Session B: **Nov 21 Mon**, 3:00 - 5:00 PM
  - Session C: **Nov 23 Wed**, 10:00 - 12:00 PM (day before exam)
  - Office location: COM3-02-33 or Zoom.
- Preferable communication: consultation sessions and Piazza.
- Don't hesitate to email me for separate appointment.
- Please browse the Piazza and post your question on the forum. I might not reply your email for repeated questions during this time.