# Lecture 10 Linear Optimization

Yingda Zhai (Dr.)

School of Computing, NUS

October 25, 2022

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

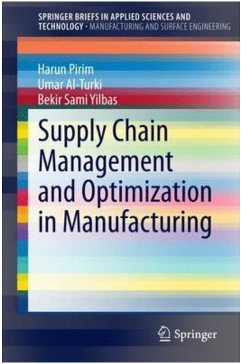
- Week 1 - 6 – Descriptive Analytics
- Week 7 – Linear Regression
- Week 8 – Logistic Reg & Time Series
- Week 9 – Data Mining Basics
- Week 10 – *Online Assessment*
- Week 11 – Linear Optimization
- Week 12 – Integer Optimization & Summary
- Week 13 – Tutorials and Consultation
- Nov 24 – *Final Exam*

*"It is not from the benevolence of the butcher, the brewer, or the baker, that we expect our dinner, but from their regard to their own interest."*

— Adam Smith, *The Wealth of Nations*

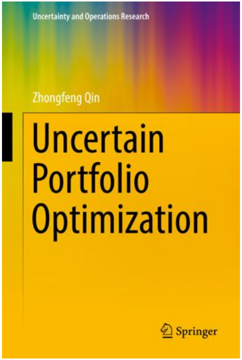
# Optimization is Everywhere

## Operation Mgt



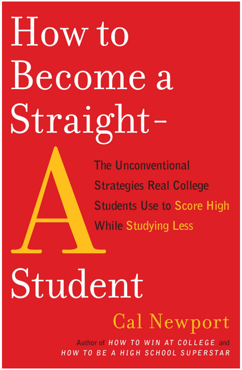

Manufacturing

## Investment




Portfolio optimization

## Career Concern




Straight A in College?

# Learning Objectives

- Be able to identify the objective function and the set of linear constraints, given a description of linear optimization problem.
- Be familiar with basic concepts in linear optimization such as feasible set and binding constraints; understand that optimal solution might arise as a corner in the feasible set.
- Be able to solve linear optimization problem with intuition and R.
- Be comfortable in interpreting the result of optimization and of sensitivity analysis and writing "prescriptive" recommendation with justification.

# Formulation of Linear Optimization Problem

# Roadmap

**Identify** — Identify decision variables, objective function and set of constraints.

**Formulate** — Formulate the linear optimization problem.

**Solve** — Solve it either manually by graph or in R.

**Compare** — Conduct sensitivity analysis.

**Interpret** — Writing recommendation and interpretation.

# Objective Function

- Our goal of linear optimization is is to maximize (or minimize) an objective function which is a linear function of both our decision variables $x$, given parameters $\theta$, i.e. $F(x; \theta)$. e.g.
  - maximize firm's profits by choosing production quantity $x$ given production technology $\theta$;
  - maximize consumer's utility/happiness by choosing quantity of consumption $x$ given preference $\theta$;
  - minimize investor's risk exposure by choosing portfolio $x$ given stock's volatility $\theta$.
- Objective such as profits, happiness, etc is optimized under some "context" captured by parameter $\theta$.
- The variables controlled by decision maker are called decision variables such as production quantity, goods to consume, stocks to buy in, etc.

## Set of Constraints

- Quite often, decision maker faces constraints to optimize her objective. For instance, we don't have unlimited budget, enough time, or need to fulfill other criteria as part of the contract, etc.
- Formally, constraints are represented as math inequalities ($\leq$), a linear constraint if the math expression is a linear inequality, e.g. $a + bx \leq c$.
- We usually put our linear expression of decision variables on the left hand and the constraint amount on the right.

| Example of constraint | Math Expression of Inequality |
|---|---|
| Produce goods within budget. | total cost $\leq$ budget |
| Produce at least 65 units of goods. | total #product $\geq$ 65 |
| Working and leisure hours daily. | workhrs + leishrs $\leq$ 24 |
| Non-negativity (often implicit) | consumption $\geq 0$, product $\geq 0$ |

- The set or region of decision variable satisfying all constraints are called feasibile set or feasible region.

# Example A: Farmer's Profit Maximization Problem

- Farmer Jean has 200 plots of lands, on which she can grow **parsnips** or **kale**.
- Jean has \$100 budget to grow.
- How many plots for each crop should she plant to maximize her profit?

| Farm: | Parsnips  | Kale  |
|---|---|---|
| Cost: | \$0.20 | \$0.70 |
| Price: | \$0.35 | \$1.10 |

- We could identify the followings:
  - Decision variables ($\boldsymbol{x}$): let $x_1$ **be the number of plots for parsnips**; and $x_2$ **the number of plots for kale**.
  - Parameters ($\theta$): 200 plots. \$100 budget, costs and prices in the table.
- What is the **objective function** ($F(\boldsymbol{x};\theta)$) for farmer Jean to maximize?

$$\texttt{Total Profit} = F(\boldsymbol{x};\theta) = (0.35 - 0.20)x_1 + (1.10 - 0.70)x_2$$
$$= 0.15x_1 + 0.40x_2$$

# Example A: Farmer's Profit Maximization Problem

- Farmer Jean has 200 plots of lands, on which she can grow **parsnips** or **kale**. Jean has \$100 budget.

| Farm: | Parsnips  | Kale  |
|---|---|---|
| Cost: | \$0.20 | \$0.70 |
| Price: | \$0.35 | \$1.10 |

- We could identify the followings:
  - Decision variables ($\boldsymbol{x}$): let $x_1$ be the number of plots for parsnips; and $x_2$ the number of plots for kale.
  - Parameters ($\theta$): 200 plots. \$100 budget, costs and prices in the table.
- What are the **set of constraints** for farmer Jean?

| Constraints | Math Expression of Inequality |
|---|---|
| Budget constraint | $0.20x_1 + 0.70x_2 \leq 100$ |
| Land constraint | $x_1 + x_2 \leq 200$ |
| Non-negative constraints | $x_1 \geq 0$ and $x_2 \geq 0$ |

# Solving Linear Optimization Problem

# Writing Down Linear Problem in One Place

- Not to confuse ourselves later, it is a good practice to formally write down our linear optimization problem: summarize all information of the problem into one system of (in-)equations to solve.

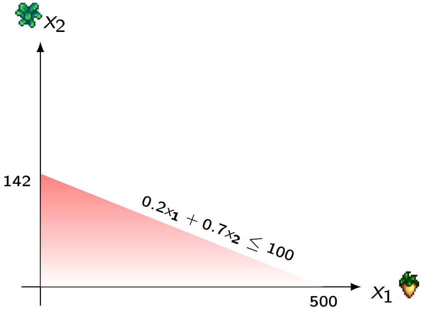
| Linear Problem | Math Expression |
|---|---|
| **Objective function:** | |
| Maximize profit choosing $\boldsymbol{x}$ | $F(\boldsymbol{x};\theta) = 0.15x_1 + 0.40x_2$ |
| **Set of constraints:** | *subject to* |
| Budget constraint | $0.20x_1 + 0.70x_2 \leq 100$ |
| Land constraint | $x_1 + x_2 \leq 200$ |
| Non-negativity | $x_1 \geq 0$ and $x_2 \geq 0$ |

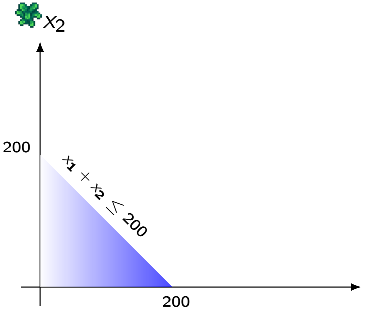
- Any solution within the feasible set is a feasible solution. Our goal is to find the **optimal** feasible solution (which maximizes profit).

# Linear Constraints in Graph

- Let's plot our constraints, linear inequalities, in $(x_1, x_2)$-plane.

| **Set of constraints** | Math Expression of Inequalities |
|---|---|
| Budget constraint | $0.20x_1 + 0.70x_2 \leq 100$ |
| Land constraint | $x_1 + x_2 \leq 200$ |
| Non-negativity | $x_1 \geq 0$ and $x_2 \geq 0$ |

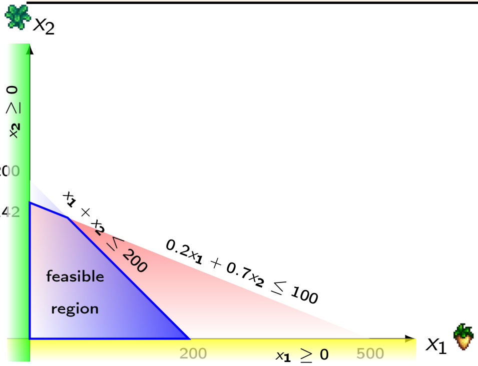


# Linear Constraints and Feasible Region in Graph

- Since optimal solution must be feasible, let's understand the problem by plotting our linear constraints as feasible region in $(x_1, x_2)$-plane.

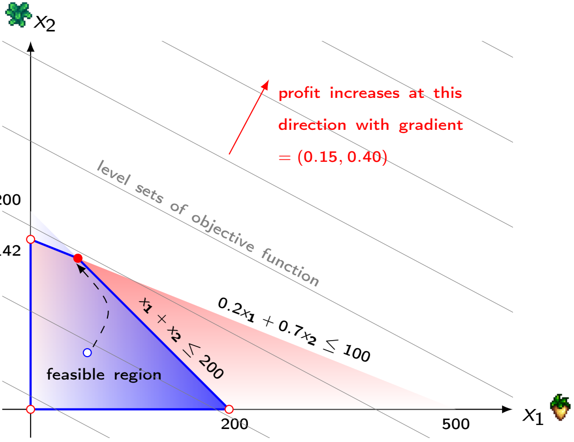
| **Set of constraints** | Math Expression of Inequalities |
|---|---|
| Budget constraint | $0.20x_1 + 0.70x_2 \leq 100$ |
| Land constraint | $x_1 + x_2 \leq 200$ |
| Non-negativity | $x_1 \geq 0$ and $x_2 \geq 0$ |




- Any feasible solution $\boldsymbol{x}$ must lie in the feasible region.
- If there is no such feasible region, no way we can could find the optimal solution. The problem becomes infeasible.

# Feasible Region and Corner Solutions

- It turns out that the optimal solution(s) for *monotone* objective function, if exists, lies at the boundary, especially a corner of the feasible region. Why?
- Level set is the set of $\boldsymbol{x}$ giving same level of profit ("contour of a mountain"). Gradient, **orthogonal** to the level sets, points to the direction where the profit increases the fastest ("steepest ridge of a mountain").



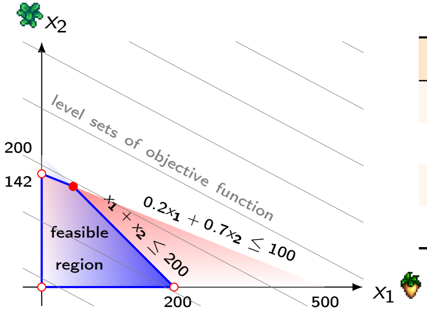

- Consider a point not on the boundary, we could move it towards the boundary following gradient ("climbing mountain").
- The optimal solution thus must be when we exhaust the constraint where we can no longer move our decision variable.
- **The optimal solution is the last point before level set leaves the feasible region.**

## Double Check on Corners

- Of course, we could check on our corner solutions by hand. Recall that:

$$\text{profit} = F(\boldsymbol{x};\theta) = 0.15x_1 + 0.40x_2$$

- First three points, $(0,0)$, $(200,0)$ and $(0,142)$ are easy to get.
- The fourth point $(x_1 = 80, x_2 = 120)$ can be obtained as intersection between two **binding constraints**, by solving $0.2x_1 + 0.7x_2 = 100$ and $x_1 + x_2 = 200$.

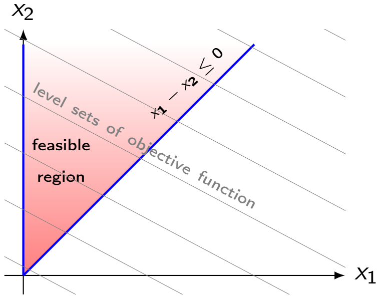


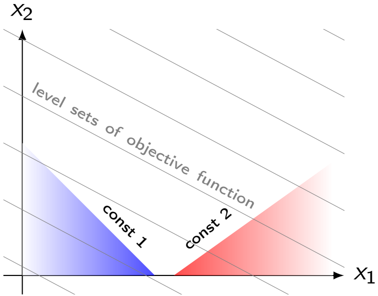

| Corners | Plug into objective function |
|---|---|
| $(0,0)$ | $0.15\cdot 0 + 0.40\cdot 0 = 0$ |
| $(200,0)$ | $0.15\cdot 200 + 0.40\cdot 0 = 30$ |
| $(0,142)$ | $0.15\cdot 0 + 0.40\cdot 142 = 56.8$ |
| $(80,120)$ | $0.15\cdot 80 + 0.40\cdot 120 = 60$ |

RM: Sometimes, there could be *a set of optimal solutions* when level set is parallel to one binding constraint.

# Four Types of Solutions

- There are four possible types of solutions in linear optimization problem:
  1. There exists a unique solution.
  2. There are multiple solutions.
  3. The solution is unbounded.
  4. There exists no feasible solution.



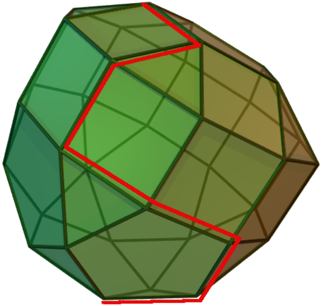




RM: It is important to identify feasible region first by plotting the constraints. Quite often we might not have any feasible solution at all in practice!

# Simplex Algorithm: Finding Corner in High Dimension

- If we have more than two decision variables, how could we systematically search all corners to locate the optimal one in high dimensional space?
- Simplex algorithm moves from one corner to another as to improve the objective function:

1. Start at a random vertex (a corner).
2. Continue along an edge ("side of polyhedron") to another vertex, only if the objective function is strictly increasing along the edge.
3. Repeat step 2 until no such edge found (no more strictly increasing edge of objective).
4. Algorithm terminates at the optimal solution, or report no solution is found, either infeasible or unbounded.



Source: simplex algorithm, wikipedia

# Solving a Linear Problem in R

- Luckily, `lpSolve::lp` in R solves such linear programs for us.

| | |
|---|---|
| Maximize profit choosing $\boldsymbol{x}$ | $F(\boldsymbol{x};\theta) = 0.15x_1 + 0.40x_2$ s.t. |
| Budget constraint | $0.20x_1 + 0.70x_2 \leq 100$ |
| Land constraint | $1x_1 + 1x_2 \leq 200$ |
| Non-negativity | $x_1 \geq 0$ and $x_2 \geq 0$ |

```
# first, define all parameters to feed 'lpSolve'
objective_function = c(0.15, 0.40)
constraint_mat = matrix(c(0.20, 0.70, 1, 1), ncol = 2, byrow = TRUE)
constraint_dir = c('<=', '<=')
constraint_rhs = c(100, 200)
```

RM:
1. `lpSolve` assumes all decision variables are non-negative.
2. `ncol` in `constraint_mat` represents the number of our decision variables.
3. Always formulate your linear problem in this way as not to confuse yourself when coding.

# Solving a Linear Problem in R

```
# then solve the linear problem using 'lp()' function
lp_solution = lp(direction = 'max', objective_function,
                 constraint_mat, constraint_dir, constraint_rhs,
                 compute.sens = TRUE)
# display the solution of linear problem: 'lp_obj$solution'
print(lp_solution$solution)
# display the value of objective function at optimal solution
print(lp_solution)
```

```
> print(lp_solution$solution)
[1]  80 120
> print(lp_solution)
Success: the objective function is 60
```

RM:
1. Good to see `lpSolve` agrees with our solution by graph!
2. Fact: max a function $F(\boldsymbol{x};\theta)$ is equivalent to min $-F(\boldsymbol{x};\theta)$.
3. `compute.sens = TRUE` gives us sensitivity analysis result.

# Example B: Vegan's Consumption Problem

- Vegan Mason has daily diet consisting of **soybeans**, **parsnips** and **kale**, which brings him happiness (utility) from consumption.

| Diet: | **Soybeans** | **Parsnips** | **Kale** |
|---|---|---|---|
| Utility: | 250 | 225 | 300 |
| Price ($): | 7 | 5 | 8 |
| Storage (L): | 15 | 30 | 40 |

- Mason has a refrigerator of capacity of 200L to keep them fresh.
- He has a weekly budget of $60 to spend.
- His doctor tells Mason not to consume more than 7 units of soybean for his high uric acid level.
- **How does Mason choose his diet profile to make himself a happiest vegan?**

# Formulation of Vegan's Utility Maximization Problem

- It is straightforward to see our **Decision variables** are:
  - $x_1$: unit of soybean consumed.
  - $x_2$: unit of parsnip consumed.
  - $x_3$: unit of kale consumed.
- We could formulate our linear problem with three decision variables:

| 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 $x_3 \geq 0$ |

RM: This is formally called consumer's utility maximization problem.

# Coding Vegan's Problem 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 | $x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0$ |

```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, compute.sens = TRUE)
```

# Solving Vegan's Problem in R

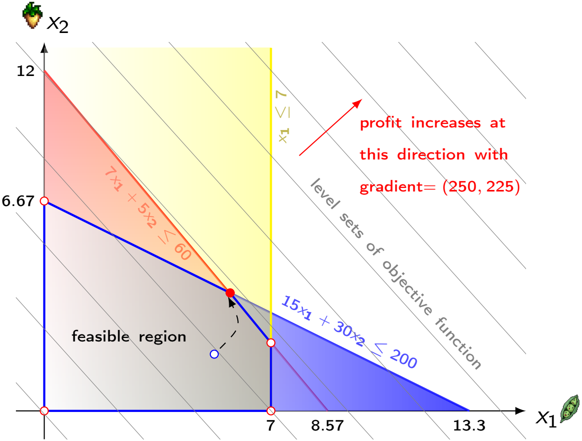
```
# then solve model
lp.solution <- lp("max", objective.fn, const.mat,
                  const.dir, const.rhs, compute.sens=TRUE)
# optimal solution for consumption profile
print(lp.solution$solution)
# objective fn value at optimal solution
print(lp.solution)
```

```
> print(lp.solution$solution)
[1] 5.925926 3.703704 0.000000
> print(lp.solution)
Success: the objective function is 2314.815
```

- Optimal solution is $x_1 = 5.93$, $x_2 = 3.70$ and $x_3 = 0.00$. Recall,
  - $x_1$: unit of soybean consumed.
  - $x_2$: unit of parsnip consumed.
  - $x_3$: unit of kale consumed.
- Therefore, it is **not optimal to consume any kale**.

# Vegan's Problem: Binding Constraints in Graph

- Let's ignore consumption of kale $x_3$ at the moment and plot constraints onto $(x_1, x_2)$-coordinates.




- The solution lies at the intersect of first $7x_1 + 5x_2 \leq 60$ and second $15x_1 + 30x_2 \leq 200$ constraints, but not third constraint $x_1 \leq 7$.
- We say the first two are binding constraints while the third is non-binding.
- The optimal solution is thus $x_1 = 5.93$, $x_2 = 3.70$ (and $x_3 = 0.00$).

RM: Any non-negativity constraint $x_1 \geq 0$, $x_2 \geq 0$ or $x_3 \geq 0$ binding?

# Sensitivity Analysis

# Comparative Statistics: Sensitivity Analysis

- We obtain an optimal solution under one set of parameter values $\theta$. However, oftentimes we are more interested in: **what if** the some parameter values $\theta$ change? ("counterfactual - prescriptive analytics")
- How would **optimal solution** change if a **parameter** changes? How **sensitive** is the *optimal solution* to changes of parameter in the problem?
- Sensitivity analysis allows us to answer this kind of questions as comparative statistics in a systematic way. We shall mainly cover two types of sensitivity analysis:

1. Changing objective function coefficients.
2. Changing value on the right hand side of a constraint. ("shadow price")

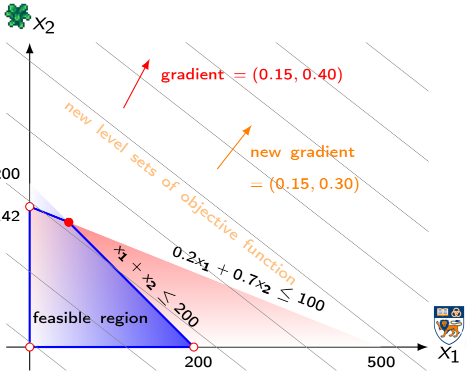
## Changes in Coefficients of Objective Function

- In Farmer Jean's problem, optimal solution is given by: $x_1 = 80$ and $x_2 = 120$.
- Total profit is $0.15x_1 + 0.30x_2$.

| Farm: | Parsnips | Kale |
|---|---|---|
| Cost: | \$0.20 | \$0.70 |
| Price: | \$0.35 | \$1.00 |

- What if market price for kale is reduced to \$1, i.e. average profit of kale reduced from 0.40 to 0.30? Does it change Jean's optimal production plan?

  - **Alternation in objective function coefficients** changes the direction of gradient (and of course, the level set).
  - In this case, optimal solution stays the same since it is still the last corner that the level set leaves the feasible region.

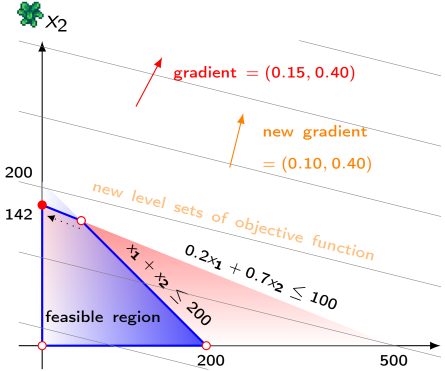


# Changes in Coefficients of Objective Function

- In Farmer Jean's problem, optimal solution is given by: $x_1 = 80$ and $x_2 = 120$.
- Total profit is $0.10x_1 + 0.40x_2$.

| Farm: | Parsnips | Kale |
|---|---|---|
| Cost: | \$0.20 | \$0.70 |
| Price: | 0.30 | \$1.10 |

- What if market price for parsnip is reduced to \$0.30, i.e. average profit of parsnip reduced from 0.15 to 0.10? Change of plan?
  - **Alternation in objective function coefficients** changes the direction of gradient (and of course, the level set).
  - In this case, optimal solution moves to another corner ($x_1 = 0$ and $x_2 = 142$), i.e. optimal to halt parsnip production.
  - **Comparing slopes of level set with those of constraints** leads to answer without any computation!



# Range of Objective Function Coefficients in R: Farmer's Problem

- How do we obtain the range of objective function coefficients where **current solution** remains optimal? Profit $= 0.15x_1 + 0.40x_2$.

```
# then solve the linear problem using 'lp()' function
lp_solution = lp(direction = 'max', objective_function,
              constraint_mat, constraint_dir, constraint_rhs,
                  compute.sens = TRUE)
# display range of objective coefs where current solution is valid
print(lp_solution$sens.coef.from); print(lp_solution$sens.coef.to);
range_objcoef = cbind(lp_solution$sens.coef.from, lp_solution$sens.coef.to)
rownames(range_objcoef) = c('x1','x2');colnames(range_objcoef) = c('from','to')
print(range_objcoef)
```

```
> print(lp_solution$sens.coef.from); print(lp_solution$sens.coef.to)
[1] 0.1142857 0.1500000
[1] 0.400 0.525
> print(range_objcoef)
       from    to
x1 0.1142857 0.400
x2 0.1500000 0.525
```

e.g. It means current solution is still optimal, as long as coef on $x_1$ lies between [0.114, 0.400]

If either coefficient is beyond the range, optimal solution will moves to another vertex.

RM: 1 These ranges assume that **only one** objective coefficient is shifted at a time.
2 Watch out how we read these numbers! The way they are listed can be confusing.

# Range of Objective Function Coefficients in R: Vegan's Problem

- Let's do the same for Vegan's diet problem: utility $= 250x_1 + 225x_2 + 300x_3$.

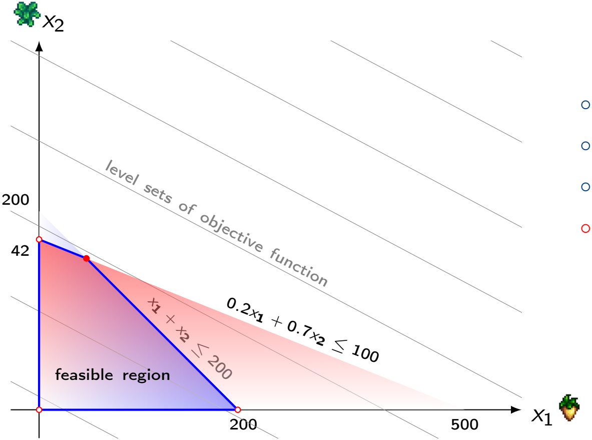
```r
# display range of objective coefs where current solution is valid
range.objcoef = cbind(lp.solution$sens.coef.from, lp.solution$sens.coef.to)
rownames(range.objcoef) = c('x1', 'x2', 'x_3')
colnames(range.objcoef) = c('from', 'to')
print(range.objcoef)
```

```
> print(range.objcoef)
         from    to
x1  1.125e+02 315.0
x2  1.906e+02 500.0
x3 -1.000e+30 340.7
```

- The current solution, $x_1 = 5.93$, $x_2 = 3.70$ and $x_3 = 0$, remains optimal as long as: (again, **only one** coefficient varies at a time)
  1. Marginal utility of soybeans lies between $[112.5, 315]$.
  2. Marginal utility of parsnip lies between $[190.6, 500]$.
  3. Marginal utility of kale lies between $(-\infty, 340.7]$.
- In order to include kale into the *optimal* diet, the marginal utility of kale must be strictly greater than 340.7!

# Shadow Price of Budget Constraint in Farmer Jean's Problem

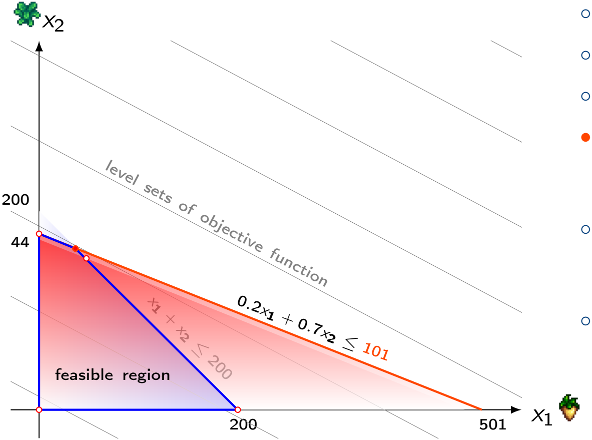
- The shadow price of a linear constraint captures the change in the **objective function's optimal value** per unit-increase in the right hand side parameter value of that constraint, holding all else equal.
- What if Jean has **one more dollar** she could spend, i.e. budget constraint is **relaxed** from 100 to 101?




- Two **binding constraints**:
- $0.20x_1 + 0.70x_2 = 100$ (budget)
- $x_1 + x_2 = 200$ (land)
- Optimal solution before was $x_1 = 80$ and $x_2 = 120$. Optimal profit is 60.

# Shadow Price of Budget Constraint in Farmer Jean's Problem

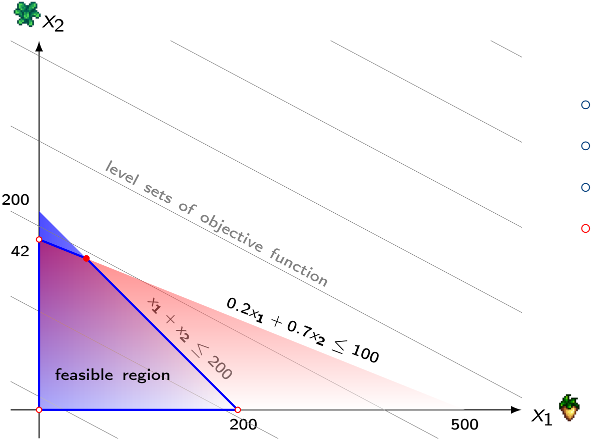
- The shadow price of a linear constraint captures the change in the **objective function** value per unit-increase in the right hand side parameter value of that constraint, holding all else equal.
- What if Jean has **one more dollar** she could spend, i.e. budget constraint is *relaxed* from 100 to 101?




- Same **binding constraints**:
- $0.20x_1 + 0.70x_2 = 101$ (budget)
- $x_1 + x_2 = 200$ (land)
- Optimal solution changes to $x_1 = 78$ and $x_2 = 122$. Optimal profit is \$60.5.
- Feasible region becomes **larger**, moving optimal solution **up and to the left**.
- Shadow price of budget constraint is thus $60.5 - 60 = \$0.5$.

# Shadow Price of Land Constraint in Farmer Jean's Problem

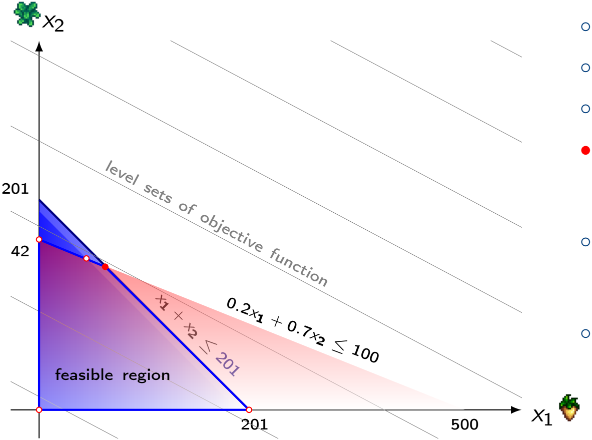
- What if Jean has **one more plot of land**, i.e. land constraint is **relaxed** from 200 to 201?




- Two **binding constraints**:
- $0.20x_1 + 0.70x_2 = 100$ (budget)
- $x_1 + x_2 = 200$ (land)
- Optimal solution before was $x_1 = 80$ and $x_2 = 120$. Optimal profit is 60.

# Shadow Price of Land Constraint in Farmer Jean's Problem

- What if Jean has **one more plot of land**, i.e. land constraint is **relaxed** from 200 to 201?




- Same **binding constraints**:
- $0.20x_1 + 0.70x_2 = 100$ (budget)
- $x_1 + x_2 = 201$ (land)
- Optimal solution changes to $x_1 = 81.4$ and $x_2 = 119.6$. Optimal profit is \$60.05.
- Feasible region becomes **larger**, moving optimal solution **down and to the right**.
- Shadow price of land constraint is thus $60.05 - 60 = \$0.05$.

RM: *Shadow price is the price of constrained resource*. If you know the math, shadow price of a constraint *is* the value of Lagrangian multiplier assigned to that constraint, which is called a dual variable, relative to original decision variables.

# Shadow Prices in R: Farmer's Problem

- Recall that we set `compute.sens = TRUE` in `lpSolve` for Farmer Jean's problem. Shadow prices can be obtained from `lp_solution$duals`.

```
# then solve the linear problem using 'lp()' function
lp_solution = lp(direction = 'max', objective_function,
                 constraint_mat, constraint_dir, constraint_rhs,
                 compute.sens = TRUE)
# display shadow prices of constraints in sensitivity analysis
print(lp_solution$duals)
```

```
> print(lp_solution$duals)
[1] 0.50 0.05 0.00 0.00
```

RM:
1. Reported order follows the order of how we coded constraints into `constraint_mat`.
2. Consistent results between program and our computation by hand.
3. The last two zeroes are shadow prices of non-negativity constraints, i.e. $x_1 \geq 0$ and $x_2 \geq 0$, which are sometimes called reduced costs. Note that they are zeros because both non-negativity constraints are non-binding at optimal solution.

# Shadow Prices in R: Vegan's Problem

- Let's use `lpSolve` to get shadow prices in Vegan's diet problem.

```
# 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, compute.sens=TRUE)
# display shadow prices of constraints
print(lp.solution$duals)
```

```
> print(lp.solution$duals)
[1]  30.555556   2.407407   0.000000   0.000000   0.000000 -40.740741
```

RM:
1. Recall the optimal solution is $x_1 = 5.93$, $x_2 = 3.70$ and $x_3 = 0$.
2. Relaxing budget constraint (one more dollar for budget) benefits the most.
3. Why is the shadow price for non-negativity constraint $x_3 \geq 0$ non-zero? Why is it negative?

## Roadmap to Linear Optimization Problem

| | |
|---|---|
| **Identify** | Identify decision variables, objective function and set of constraints. |
| **Formulate** | Formulate the linear optimization problem. |
| **Solve** | Solve it either manually by graph or in R. |
| **Compare** | Conduct sensitivity analysis. |
| **Interpret** | Writing recommendation and interpretation. |

# Recommendation Coda for Farmer Jean (1)

- Jean is still waiting for our help to optimize the field production.
- How to write a recommendation memo to summarize our prescriptive analysis?

**Identify**

State clearly our objective/goal, decision variables and constraints of the problem.

**Formulate**

In practice, problem formulation might not be necessarily included into memo if Jean (client) does not understand optimization. Yet necessary for our own purpose.

**Solve**

With R solver, "`Optimal planting scheme is parsnip = 80 and kale = 120, which gives you profit of 60.`" Graphing problem is strongly recommended since it provides key insight about which constraint is binding/non-binding, i.e. what holds Jean back to become a better farmer.

# Recommendation Coda for Farmer Jean (II)

- Jean is still waiting for our help to optimize the field production.
- How to write a recommendation memo to summarize our prescriptive analysis?

**Compare**

- Examine the range of coefficients over which the current solution is valid.

e.g. "`If the average profit of kale goes above $0.53 or that of parsnip goes below $0.11, please consider changing your production plan.`"

- Examine and interpret shadow prices (relaxing constraints).

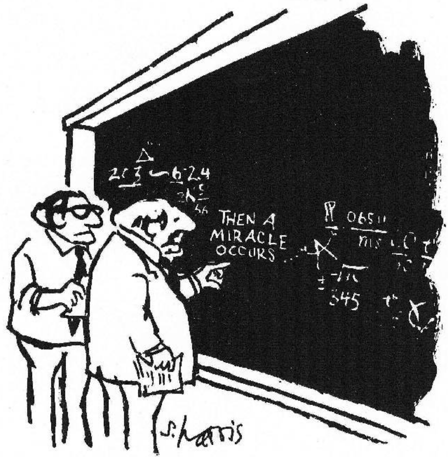
e.g. "`We recommend you to try to relax your budget since one more dollar of budget to purchase seeds increases your profit by $0.50 (by changing planting scheme to ...) while getting one more plot of land only increases your profit by $0.05.`" (It might be much more costly for Jean to relax land constraint than budget.)

**Interpret**

Thank Jean and tell her to expect your invoice in the mail. :D

# Summary

- In today's lecture, we discussed how to formulate and solve linear optimization problems, which offers us a "prescription" to optimize business decision.
- Given the complexity of the real world and many advanced optimization theories and algorithms, many real-world problems, surprisingly, can be formulated or approximated by linear programming, which can be solved efficiently to provide useful business intelligence.
- We assume all our choice variables are *continuous* and taking values from real line $\mathbb{R}$.
- Next week, we shall round off our module with more on linear optimization where some choice variables need to be **integers** or even **binary** i.e. yes/no decisions.




"I think you should be more explicit here in step two."