

Data Science for Actuaries (ACT6100)

Arthur Charpentier

Rappels # 4.5 (Linear Programming)

automne 2020

 <https://github.com/freakonometrics/ACT6100/>

Linear Programming

The beer problem: we want to produce beer, either blonde, or brown. We produce q_{blond} and q_{brown} barrels.



{ hops : 5kg
barley : 10kg
price : 40\$



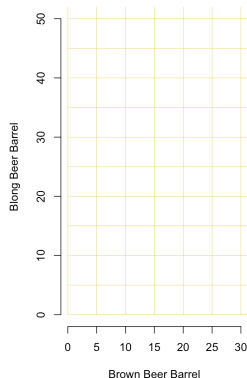
{ hops : 2kg
barley : 14kg
price : 30\$



{ hops : 100kg
barley : 280kg

We want to find $q_{\text{brown}} (x)$ and $q_{\text{blond}} (y)$
that is feasible and maximize profit...

(1) (2)



Linear Programming

The beer problem: we want to produce beer, either blonde, or brown. We produce q_{blond} and q_{brown} barrels.



$\left\{ \begin{array}{l} \text{hops : } 5\text{kg} \\ \text{barley : } 10\text{kg} \\ \text{price : } 40\$ \end{array} \right.$

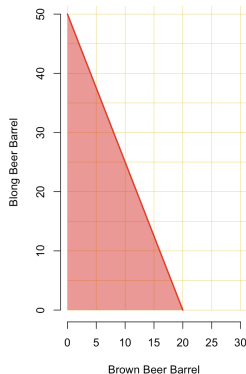


$\left\{ \begin{array}{l} \text{hops : } 2\text{kg} \\ \text{barley : } 14\text{kg} \\ \text{price : } 30\$ \end{array} \right.$



$\left\{ \begin{array}{l} \text{hops : } 100\text{kg} \\ \text{barley : } 280\text{kg} \end{array} \right.$

(1) Our **hops** stock is 100kg,
it takes 5kg per barrel of blonde
it takes 2kg per barrel of brown
so $5x + 2y \leq 100$



Linear Programming

The beer problem: we want to produce beer, either blonde, or brown. We produce q_{blond} and q_{brown} barrels.



$\left\{ \begin{array}{l} \text{hops : } 5\text{kg} \\ \text{barley : } 10\text{kg} \\ \text{price : } 40\$ \end{array} \right.$

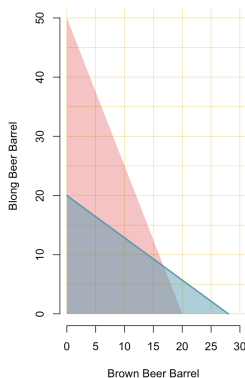


$\left\{ \begin{array}{l} \text{hops : } 2\text{kg} \\ \text{barley : } 14\text{kg} \\ \text{price : } 30\$ \end{array} \right.$



$\left\{ \begin{array}{l} \text{hops : } 100\text{kg} \\ \text{barley : } 280\text{kg} \end{array} \right.$

(1) Our **barley** stock is 280kg,
it takes 10kg per barrel of blonde
it takes 14kg per barrel of brown
so $10x + 14y \leq 280$



Linear Programming

The beer problem: we want to produce beer, either blonde, or brown. We produce q_{blond} and q_{brown} barrels.



$\left\{ \begin{array}{l} \text{hops : } 5\text{kg} \\ \text{barley : } 10\text{kg} \\ \text{price : } 40\$ \end{array} \right.$



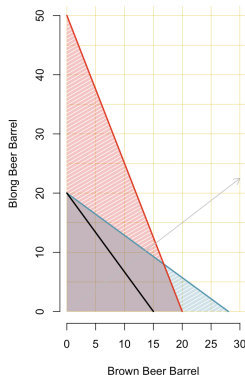
$\left\{ \begin{array}{l} \text{hops : } 2\text{kg} \\ \text{barley : } 14\text{kg} \\ \text{price : } 30\$ \end{array} \right.$



$\left\{ \begin{array}{l} \text{hops : } 100\text{kg} \\ \text{barley : } 280\text{kg} \end{array} \right.$

Under the two feasibility conditions,

(2) we want to maximize our profit
 $\max\{40x + 30y\}$



Linear Programming

The beer problem: we want to produce beer, either blonde, or brown. We produce q_{blond} and q_{brown} barrels.



$$\begin{cases} \text{hops :} & 5\text{kg} \\ \text{barley :} & 10\text{kg} \\ \text{price :} & 40\$ \end{cases}$$



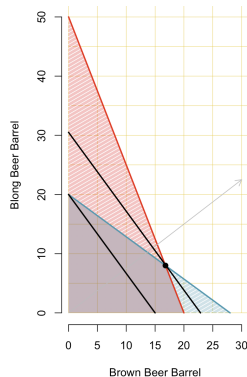
$$\begin{cases} \text{hops :} & 2\text{kg} \\ \text{barley :} & 14\text{kg} \\ \text{price :} & 30\$ \end{cases}$$



$$\begin{cases} \text{hops :} & 100\text{kg} \\ \text{barley :} & 280\text{kg} \end{cases}$$

Our problem is

$$\begin{aligned} & \max \{40x + 30y\} \\ & \text{s.t. } 10x + 14y \leq 280 \\ & \quad 5x + 2y \leq 100 \\ & \quad x, y \geq 0 \end{aligned}$$



Linear Programming

Our problem is here

$$\begin{aligned} \max \{ &\mathbf{c}^\top \mathbf{x} \} \\ \text{s.t. } &\mathbf{Ax} \leq \mathbf{b} \\ &\mathbf{x} \geq 0 \end{aligned}$$

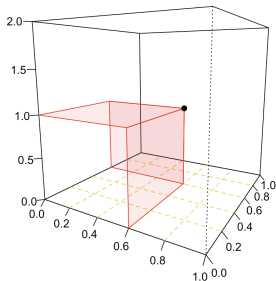
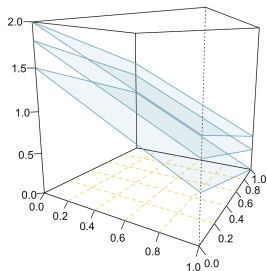
e.g.

$$\max \{ ax + by + cz \}$$

subject to

$$\begin{cases} x \leq \alpha \\ y \leq \beta \\ z \leq \gamma \end{cases}$$

The red volume is the set of feasible points
(x, y, z)



Linear Programming

Our problem is here

$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

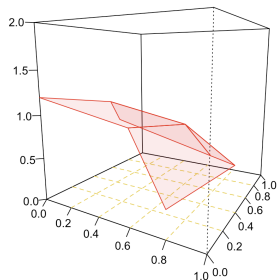
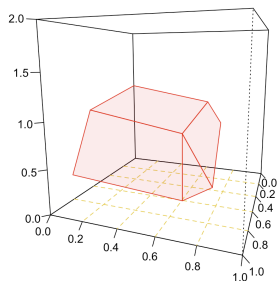
e.g., more generally

$$\max \{ax + by + cz\}$$

subject to

$$\left\{ \begin{array}{l} \alpha_1 x + \beta_1 y + \gamma_1 z \leq \delta_1 \\ \alpha_2 x + \beta_2 y + \gamma_2 z \leq \delta_2 \\ \vdots \\ \alpha_k x + \beta_k y + \gamma_k z \leq \delta_k \end{array} \right.$$

which is a (convex) polyhedron.



Simplex

Our problem is here

$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

First step: enlarge the parameter space, $10x_1 + 14x_2 \leq 280$ becomes $10x_1 + 14x_2 - u_1 = 280$ (so called slack variables)

$$\begin{aligned} \max \quad & \{40x_1 + 30x_2\} \\ \text{s.t.} \quad & 10x_1 + 14x_2 + u_1 = 280 \\ & 2x_1 + 5x_2 + u_2 = 100 \\ & x_1, x_2, u_1, u_2 \geq 0 \end{aligned}$$

which is a problem of the general form

$$\begin{aligned} \max \quad & \mathbf{c}^\top \mathbf{x} \\ \text{s.t.} \quad & \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

Simplex Method

A linear program in standard form can be written

$$\begin{aligned} & \max \{ \mathbf{c}^\top \mathbf{x} \} \\ & \text{s.t. } \mathbf{Ax} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned}$$

and it can be represented as a **tableau** (matrix) of the form

$$\begin{bmatrix} \mathbf{1} & -\mathbf{c}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{A} & \mathbf{b} \end{bmatrix}$$

Use some linear algebra, one can solve the optimization problem

Application to median computation

$\mathbf{y} = \{y_1, \dots, y_n\}$, the median is a solution to $\min_{\mu} \left\{ \sum_{i=1}^n |y_i - \mu| \right\}$.

Equivalently, we want to solve

$$\min_{\mu, \mathbf{a}, \mathbf{b}} \left\{ \sum_{i=1}^n a_i + b_i \right\}$$

with $a_i, b_i \geq 0$ and $y_i - \mu = a_i - b_i, \forall i = 1, \dots, n$.

Heuristically, the idea is to write $y_i = \mu + \varepsilon_i$, and then define a_i 's and b_i 's so that $\varepsilon_i = a_i - b_i$ and $|\varepsilon_i| = a_i + b_i$, i.e.

$$a_i = (\varepsilon_i)_+ = \max\{0, \varepsilon_i\} = |\varepsilon_i| \cdot \mathbf{1}_{\varepsilon_i > 0}$$

and

$$b_i = (-\varepsilon_i)_+ = \max\{0, -\varepsilon_i\} = |\varepsilon_i| \cdot \mathbf{1}_{\varepsilon_i < 0}$$

Application to median computation

Thus, set $\mathbf{z} = (\mu^+; \mu^-; \mathbf{a}, \mathbf{b})^\top \in \mathbb{R}_+^{2n+2}$, and then write the constraint as $\mathbf{A}\mathbf{z} = \mathbf{b}$ with $\mathbf{b} = \mathbf{y}$ and $\mathbf{A} = [\mathbf{1}_n; -\mathbf{1}_n; \mathbb{I}_n; -\mathbb{I}_n]$. For the objective function $\mathbf{c} = (\mathbf{0}, \mathbf{1}_n, -\mathbf{1}_n)^\top \in \mathbb{R}_+^{2n+2}$ and our program is $\min_{\mathbf{z}} \{\mathbf{c}^\top \mathbf{z}\}$ s.t. $\mathbf{A}\mathbf{z} = \mathbf{b}, \mathbf{z} \geq \mathbf{0}$.

```
1 > n = 101
2 > set.seed(1)
3 > y = rlnorm(n)
4 > median(y)
5 [1] 1.077415
6 >
7 > library(lpSolve)

1 > X = rep(1,n)
2 > A = cbind(X, -X, diag(n), -diag(n))
3 > b = y
4 > c = c(rep(0,2), rep(1,n), rep(1,n))
5 > r = lp("min", c, A, rep("=", n), b)
6 > head(r$solution,1)
7 [1] 1.077415
```

since the median is a solution to

$$\min_{\mu} \left\{ \sum_{i=1}^n |y_i - \mu| \right\} = \min_{\mu} \left\{ \sum_{i=1}^n \max\{(y_i - \mu), -(y_i - \mu)\} \right\}$$

Application to quantile computation

More generally, if the quantile of order τ is a solution of $\tau \in (0, 1)$,

$$\min_q \left\{ \sum_{i=1}^n \max\{\tau(y_i - \mu), (1 - \tau)(y_i - \mu)\} \right\}$$

The linear program is now

$$\min_{q^+, q^-, a, b} \left\{ \sum_{i=1}^n \tau a_i + (1 - \tau) b_i \right\}$$

with $a_i, b_i, q^+, q^- \geq 0$ and $y_i = q^+ - q^- + a_i - b_i, \forall i = 1, \dots, n$.

1	> tau = .3	1	> c = c(rep(0,2), tau*rep(1,n), (1-tau)*rep(1,n))
2	> quantile(y,tau)	2	> r = lp("min", c, A, rep("=",n), b)
3	30%	3	> head(r\$solution,1)
4	0.6741586	4	[1] 0.6741586

Application to quantile regression

In a regression, we use $\mathbf{x}_i^\top \boldsymbol{\beta}$ instead of μ . The linear program is

$$\min_{\boldsymbol{\beta}^+, \boldsymbol{\beta}^-, \mathbf{a}, \mathbf{b}} \left\{ \sum_{i=1}^n \tau a_i + (1 - \tau) b_i \right\}$$

with $a_i, b_i \geq 0$ and $y_i = \mathbf{x}_i^\top [\boldsymbol{\beta}^+ - \boldsymbol{\beta}^-] + a_i - b_i, \forall i = 1, \dots, n$ and $\beta_j^+, \beta_j^- \geq 0 \forall j = 0, \dots, k$.

```
1 > n=nrow(Davis)
2 > X = cbind( 1, Davis$height)
3 > y =Davis$weight
4 > K = ncol(X)
5 > N = nrow(X)
6 > A = cbind(X,-X,diag(N),-diag(N))
7 > c =c(rep(0,2*ncol(X)),tau*rep(1,N),(1-tau)*rep(1,N))
8 > b = y
9 > r = lp("min",c,A,rep("=",N),b)
10 > beta = r$sol[1:K] - r$sol[(1:K+K)]
11 > beta
12 [1] -110      1
```

Application to quantile regression

See

```
1 > library(quantreg)
2 > reg=rq(weight~height,data=Davis,tau=tau)
3 > summary(reg)
4
5 tau: [1] 0.3
6
7 Coefficients:
8             coefficients      lower bd      upper bd
9 (Intercept) -110.00000      -135.94169      -70.39413
10 height          1.00000          0.73202          1.17531
```

to compare with

```
1 > reg=(lm(weight~height,data=Davis))
2 > cbind(reg$coefficient,confint(reg))
3             2.5 %      97.5 %
4 (Intercept) -130.910400 -153.643662 -108.177137
5 height          1.150092   1.016991   1.283192
```