คณะสถิติประยุกต์
Graduate School of Applied Statistics

INTRODUCTION TO R FOR DECISION MAKING
OPTIMIZATION & SIMULATION

Kannapha Amaruchkul

Graduate School of Applied Statistics

National Institute of Development Administration (NIDA), Bangkok, Thailand

Friday April 29, 2016

National University of Laos (NUOL), Vientiane,

Laos PDR

Outline

- Introduction to Optimization
  - Motivating example: Product mix model
  - Sensitivity analysis
  - Examples
    - Transportation model
    - Chance constrained programming
- Introduction to Monte Carlo Simulation
  - Motivating example: Newsvendor model
  - Performance measures
  - Examples
    - Queuing model
    - Correlated inputs

Motivating Example
Simple Product Mix Model

Resource\Product	Table1	Table2	Resource Availability
A	20	40	400
B	8	32	256
C	30	20	440
Profit	230	180	

Let x_i be the number of type-i tables to be produced; $i = 1, 2$

Maximize $\quad z = 230\,x_1 + 180\,x_2$

Subject to

$$20\,x_1 + 40\,x_2 \leq 400$$
$$8\,x_1 + 32\,x_2 \leq 256$$
$$30\,x_1 + 20\,x_2 \leq 440$$
$$x_1, x_2 \geq 0$$

Excel Solver

	A	B	C	D	E
1		Table1	Table2		
2	Profit	230	180	Total profit	
3	Number to produce	12	4	3480	
4					
5	Resource\Product	Table1	Table2	Resource Used	Resource Availability
6	A	20	40	400	400
7	B	8	32	224	256
8	C	30	20	440	440

Graphical solution

A solution $(x_1, x_2) = (4,5)$ is a feasible solution, and the corresponding objective function is 1820.

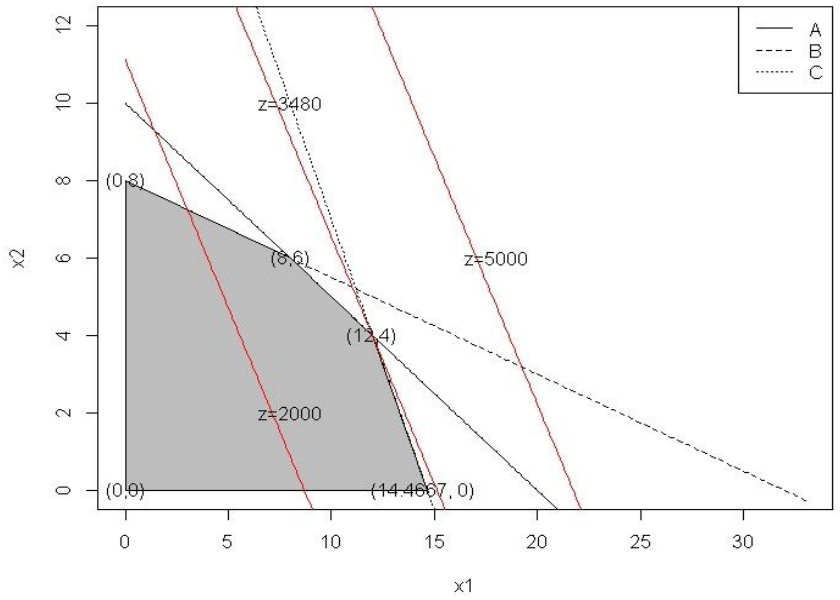
```
> profit <- c(230, 180)
> cap <- c(400, 256, 440)
> Ar <- cbind(c(20, 8, 30),
          c(40, 32, 20))
> Ar
     [,1] [,2]
[1,]   20   40
[2,]    8   32
[3,]   30   20

> x1 <- c(4,5)
> Ar%*%x1
     [,1]
[1,]  280
[2,]  192
[3,]  220
> Ar%*%x1 <= cap
     [,1]
[1,] TRUE
[2,] TRUE
[3,] TRUE

> sum(profit*x1)
[1] 1820
```

Maximize

$$z = 230\,x_1 + 180\,x_2$$

Subject to

$$20\,x_1 + 40\,x_2 \leq 400$$
$$8\,x_1 + 32\,x_2 \leq 256$$
$$30\,x_1 + 20\,x_2 \leq 440$$
$$x_1, x_2 \geq 0$$



library(lpSolve)

```
lp (direction = "min", objective.in, const.mat, const.dir, const.rhs,
        transpose.constraints = TRUE, int.vec, presolve=0, compute.sens=0,
        binary.vec, all.int=FALSE, all.bin=FALSE, scale = 196, dense.const,
        num.bin.solns=1, use.rw=FALSE)
```

Arguments

direction	Character string giving direction of optimization: "min" (default) or "max."
objective.in	Numeric vector of coefficients of objective function
const.mat	Matrix of numeric constraint coefficients, one row per constraint, one column per variable (unless transpose.constraints = FALSE; see below).
const.dir	Vector of character strings giving the direction of the constraint: each value should be one of "<," "<=," "=," "==," ">," or ">=". (In each pair the two values are identical.)
const.rhs	Vector of numeric values for the right-hand sides of the constraints.

library(lpSolve): lp

- Maximize

$$z = 230\,x_1 + 180\,x_2$$

- Subject to

$$\begin{aligned} 20\,x_1 + 40\,x_2 &\leq 400 \\ 8\,x_1 + 32\,x_2 &\leq 256 \\ 30\,x_1 + 20\,x_2 &\leq 440 \\ x_1, x_2 &\geq 0 \end{aligned}$$

- Maximize $[230\ 180]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
- Subject to

$$\begin{bmatrix} 20 & 40 \\ 8 & 32 \\ 30 & 20 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{matrix} \leq \\ \leq \\ \leq \\ \geq \\ \geq \end{matrix} \begin{bmatrix} 400 \\ 256 \\ 440 \\ 0 \\ 0 \end{bmatrix}$$

```
profit <- c(230, 180)
cap <- c(400, 256, 440)
Ar <- cbind(c(20, 8, 30), c(40, 32, 20))
con.mat <- rbind(Ar, diag(2))
con.dir <- c(rep("<=", 3), rep(">=", 2))
rhs <- c(cap, 0,0)
```

```
> library(lpSolve)

> mylp <- lp(direction="max",
          objective.in=profit,
          const.mat=con.mat,
          const.dir=con.dir,
          const.rhs=rhs)

> mylp$objval
[1] 3480
> mylp$solution
[1] 12  4
```

Sensitivity analysis

```
> mylp <- lp("max", profit, con.mat, con.dir, rhs, compute.sens=1)
> mylp$duals
[1] 1 0 7 0 0 0 0

> mylp$duals.from
[1]  2.933e+02 -1.0e+30  3.600e+02 -1.0e+30 -1.0e+30 -1.0e+30 -
1.0e+30
> mylp$duals.to
[1] 4.32e+02 1.00e+30 6.00e+02 1.00e+30 1.00e+30 1.00e+30 1.00e+30
```

Variable Cells

Cell	Name	Final Value	Reduced Cost	Objective Coefficient	Allowable Increase	Allowable Decrease
$B$2	Number to produce Table1	12	0	230	40	140
$C$2	Number to produce Table2	4	0	180	280	26.66666667

Constraints

Cell	Name	Final Value	Shadow Price	Constraint R.H. Side	Allowable Increase	Allowable Decrease
$D$6	A Resource Used	400	1	400	32	106.6666667
$D$7	B Resource Used	224	0	256	1E+30	32
$D$8	C Resource Used	440	7	440	160	80

library(Rglpk)

```
Rglpk_solve_LP(obj, mat, dir, rhs, bounds = NULL, types = NULL, max = FALSE,
               control = list(), ...)
```

Arguments

obj	a numeric vector representing the objective coefficients.
mat	a numeric vector or a matrix of constraint coefficients.
dir	a character vector with the directions of the constraints. Each element must be one of "<", "<=", ">", ">=", or "==".
rhs	the right hand side of the constraints.
bounds	NULL (default) or a list with elements upper and lower containing the indices and corresponding bounds of the objective variables. The default for each variable is a bound between 0 and Inf.
types	a character vector indicating the types of the objective variables. types can be either "B" for binary, "C" for continuous or "I" for integer. By default NULL, taken as all-continuous. Recycled as needed.
max	a logical giving the direction of the optimization. TRUE means that the objective is to maximize the objective function, FALSE (default) means to minimize it.

Value

A list containing the optimal solution, with the following components.

solution	the vector of optimal coefficients
objval	the value of the objective function at the optimum
status	an integer with status information about the solution returned. If the control parameter canonicalize_status is set (the default) then it will return 0 for the optimal solution being found, and non-zero otherwise. If the control parameter is set to FALSE it will return the GLPK status codes.

library(Rglpk): Rglpk_solve_LP

- Maximize

$$z = 230\,x_1 + 180\,x_2$$

- Subject to

$$\begin{aligned} 20\,x_1 + 40\,x_2 &\leq 400 \\ 8\,x_1 + 32\,x_2 &\leq 256 \\ 30\,x_1 + 20\,x_2 &\leq 440 \\ x_1, x_2 &\geq 0 \end{aligned}$$

- Maximize $[230\ 180]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$
- Subject to

$$\begin{bmatrix} 20 & 40 \\ 8 & 32 \\ 30 & 20 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{matrix} \leq \\ \leq \\ \leq \\ \geq \\ \geq \end{matrix} \begin{bmatrix} 400 \\ 256 \\ 440 \\ 0 \\ 0 \end{bmatrix}$$

```
profit <- c(230, 180)
cap <- c(400, 256, 440)
Ar <- cbind(c(20, 8, 30), c(40, 32, 20))
con.mat <- rbind(Ar, diag(2))
con.dir <- c("<=", "<=", "<=", ">=", ">=")
rhs <- c(cap, 0,0)
```

```
> mylp2 <-Rglpk_solve_LP(max=TRUE,
          obj=profit,
          mat=Ar,
          dir=rep("<=", 3),
          rhs=cap)

> mylp2$optimum
[1] 3480
> mylp2$solution
[1] 12  4
```

Transportation model

	Unit transportation cost				Capacity
Plant\Warehouse	1	2	3	4	
1	4.60	5.00	5.43	8.65	70
2	3.80	4.20	7.90	7.81	130
3	7.58	6.97	4.12	6.75	100
Demand	85	75	60	80	

```
costmx <- matrix(c(4.60, 5.00, 5.43, 8.65,
                   3.80, 4.20, 7.90, 7.81,
                   7.58, 6.97, 4.12, 6.75),
         nrow=3, byrow=T)

demand <- c(85, 75, 60, 80)
pcap <- c(70, 130, 100)

myshipping <- lp.transport(costmx,
                   "min",
                   row.signs=rep("=", 3), row.rhs=pcap,
                   col.signs=rep("=",4), col.rhs=demand)

myshipping$solution
```

	A	B	C	D	E	F	G
1		Unit transportation cost					
2	Plant\Warehouse	1	2	3	4		
3	1	4.6	5	5.43	8.65		
4	2	3.8	4.2	7.9	7.81		
5	3	7.58	6.97	4.12	6.75		
6	Plant\Warehouse	1	2	3	4	Total from Plant	Capacity
7	1	0	30	40	0	70	70
8	2	85	45	0	0	130	130
9	3	0	0	20	80	100	100
10	Total to WH	85	75	60	80		
11	Demand	85	75	60	80		
12	Total cost	1501.6					

```
B12 =SUMPRODUCT(B3:E5,B7:E9)
```

	A	B	C	D	E	F	G	H
1	Origin (Plant)	Destination (W/H)	Cost	Flow		Plant Cons.	Flow from plant	Capacity
2	1	1	4.6	0		1	70	70
3	1	2	5	30		2	130	130
4	1	3	5.43	40		3	100	100
5	1	4	8.65	0		WH Cons.	Flow to WH	Demand
6	2	1	3.8	85		1	85	85
7	2	2	4.2	45		2	75	75
8	2	3	7.9	0		3	60	60
9	2	4	7.81	0		4	80	80
10	3	1	7.58	0				
11	3	2	6.97	0				
12	3	3	4.12	20				
13	3	4	6.75	80				
14	Total cost			1501.6				

```
G2 =SUMIF($A$2:$A$13,F2,$D$2:$D$13)
G6 =SUMIF($B$2:$B$13,F6,$D$2:$D$13)
```

library(lpSolve): lp.transport

```
lp.transport (cost.mat, direction="min", row.signs, row.rhs, col.signs,
              col.rhs, presolve=0, compute.sens=0, integers = 1:(nc*nr) )
```

Arguments

cost.mat	Matrix of costs; ij-th element is the cost of transporting one item from source i to destination j.
direction	Character, length 1: "min" or "max"
row.signs	Vector of character strings giving the direction of the row constraints: each value should be one of "<," "<=," "=," "==," ">," or ">=." (In each pair the two values are identical.)
row.rhs	Vector of numeric values for the right-hand sides of the row constraints.
col.signs	Vector of character strings giving the direction of the column constraints: each value should be one of "<," "<=," "=," "==," ">," or ">=."
col.rhs	Vector of numeric values for the right-hand sides of the column constraints.
presolve	Numeric: presolve? Default 0 (no); any non-zero value means "yes." Currently ignored.
compute.sens	Numeric: compute sensitivity? Default 0 (no); any non-zero value means "yes."
integers	Vector of integers whose ith element gives the index of the ith integer variable. Its length will be the number of integer variables. Default: all variables are integer. Set to NULL to have no variables be integer.

Mathematical programming models

- Linear programming (LP): `lpSolve, Rglpk`
  - Transportation models: `lp.transport`
  - Transshipment models
- Integer programming (IP)
  - Binary programming: `lp(…,int.vec=1:3)`
  - Mixed-integer linear programming (MILP)
- Nonlinear programming (NLP): `Rsolnp, optim`
  - Mixed-integer nonlinear programming (MINLP)
  - Quadratic programming (QP): `quadprog`
- Multi-objective and goal programming: `goalprog, MCO`

Chance constrained programming

Product Type (i)		1	2	3	4	Resource availability
Maximum demand (u_i)		1000	2000	500	1000	
Unit profit		6.5	2.4	3.8	3.2	
Raw material A		4	2	1	2	6000
Raw material B		6	2	1	2	10000
Per unit labor (hours)	Mean (μ_i)	2	1	3	2	4000
	Stdev (σ_i)	0.3	0.2	0.15	0.75	

- Let x_i be the weekly production of type-i product
- Let T_i be labor required to produce one unit of type-i product
- Labor constraint

$$P(x_1T_1 + x_2T_2 + x_3T_3 + x_4T_4 \leq 4000) \geq 0.95$$

- Assume $T_i \sim N(\mu_i, \sigma_i^2)$

$$\sum_{i=1}^{n} \mu_i x_i + \Phi^{-1}(0.95)\sqrt{\sum_{i=1}^{n} \sigma_i^2 x_i^2} \leq 4000$$

Maximize $6.5x_1 + 2.4x_2 + 3.8x_3 + 3.2\,x_4$

Subject to

$$\sum_{i=1}^{4} \sigma_i^2 x_i^2 - y^2 = 0$$

$$2x_1 + x_2 + 3\,x_3 + 2\,x_4 + \Phi^{-1}(0.95)y \leq 4000$$

$$4x_1 + 2\,x_2 + x_3 + 2\,x_4 \leq 6000$$

$$6x_1 + 2x_2 + x_3 + 2\,x_4 \leq 1000$$

$$0 \leq x_i \leq u_i \text{ for each } i = 1,2,3,4$$

$$y \geq 0$$

library(Rsolnp): gosolnp

```
n <- 5                          # of decision variables
m <- 3                          # of constraints
A <- matrix(c(2,1,3,2,
4,2,1,2,
6,2,1,2), nrow=3, byrow=T)

alpha <- 0.95
A2 <- cbind(A, c(qnorm(alpha),0,0))
fncon <- function(x){ A2%*%x }
rhs <- c(4000, 6000, 10000)

obj <- c(6.5, 2.4, 3.8, 3.2,0)
fn <- function(x) {-1*sum(obj*x)}

sigma <- c(0.3, 0.2, 0.15, 0.75)
fn2 <- function(x){sum((x[1:(n-1)]^2)*(sigma^2))-x[n]^2}
UBx <- c(1000, 2000, 500, 1000)
UB <- c(UBx,  sum((sigma^2)*UBx))

myop <- gosolnp(fun=fn, eqfun=fn2, eqB=0,
                ineqfun=fncon, ineqLB=rep(0,m), ineqUB=rhs,
                UB=UB, LB=rep(0,n))
```

Maximize $6.5x_1 + 2.4x_2 + 3.8x_3 + 3.2\,x_4$

Subject to

$$\sum_{i=1}^{4} \sigma_i^2 x_i^2 - y^2 = 0$$

$$2x_1 + x_2 + 3\,x_3 + 2\,x_4 + \Phi^{-1}(0.95)y \le 4000$$

$$4x_1 + 2\,x_2 + x_3 + 2\,x_4 \le 6000$$

$$6x_1 + 2x_2 + x_3 + 2\,x_4 \le 1000$$

$$0 \le x_i \le u_i \text{ for each } i = 1,2,3,4$$

$$y \ge 0$$

```
> myop <- gosolnp(fun=fn, eqfun=fn2, eqB=0,
ineqfun=fncon, ineqLB=rep(0,m), ineqUB=rhs,
+ UB=UB, LB=rep(0,n))
Iter: 1 fn: -9172.4203    Pars:  973.19799 764.53053
Iter: 2 fn: -9210.1599    Pars:  996.78572 722.55118 …
Iter: 15 fn: -9337.1492   Pars:  1000.00000000
solnp--> Completed in 15 iterations
> myop$pars
[1] 1.000000e+03 9.159328e+02 1.681343e+02 1.282136e-05 3.524108e+02
> myop$values
 [1] -8517.128 -9172.420 -9210.160 -9217.340 -9217.961 -9218.102 -9218.134
 [8] -9218.142 -9218.144 -9218.144 -9218.144 -9406.327 -9332.730 -9337.133
[15] -9337.149 -9337.149
```

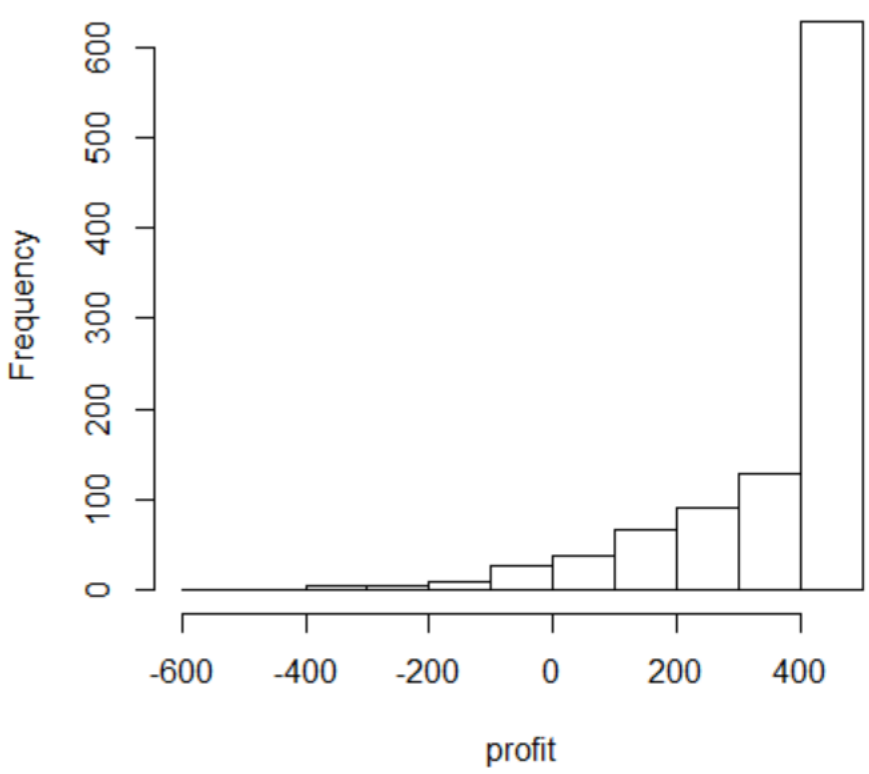
Motivating example: Newsvendor model

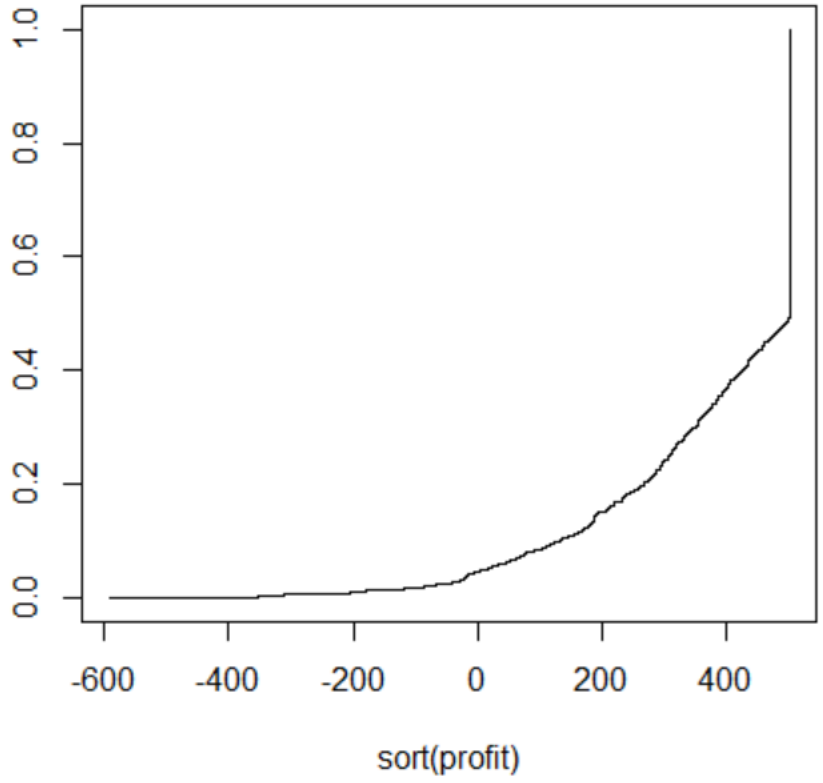
	A	B	C	D	E	F	G	H	I	J
1	**Newsvendor Model**				Pseudo random #		B16	=RAND()		
2					Normal demand		C16	=NORM.INV(B16,$F$8,$H$8)		
3	**Cost data**				Sales		D16	=MIN(C16,$B$12)		
4	Unit cost	$7.50			Leftovers		E16	=$B$12-D16		
5	Unit price	$10.00			Revenue		F16	=$B$5*D16		
6	Unit refund	$2.50			Refund		G16	=$B$6*E16		
7					Profit		H16	=F16-$B$4*$B$12+G16		
8	**Uncertain quantity: Demand is normal with**				**Mean**	**200**	**Stdev**	**40**		
9				**Performance measures**						
10				Average profit			95% confidence interval for expected profit			
11	**Decision variable**			Stdev of profit			Lower limit			
12	Order quantity	200		Prob(Profit at least 200)			Upper limit			
13				Prob(Negative profit)						
14	**Profit model**									
15	Rep#	Rand()	Demand	Sales	Leftovers	Revenue	Refund	Profit		
16	1	0.956	268	200	0	2,000	-	500		
17	2	0.231	171	171	29	1,706	74	279		

```
nRep <- 1000
demand <- rnorm(nRep, 200, 40)
OQ <- 200  #order quantity
sales <- pmin(demand, OQ)
left <- OQ-sales
profit <- 10*sales - 7.5*OQ + 2.5*left
```

Performance measures

```
> hist(profit)
> plot(sort(profit), (1:nRep)/nRep, type="s", ylim=c(0,1))
```





Distribution in R

- `rnorm` random variable
- `pnorm` CDF
- `qnorm` quantile, inverse CDF
- `dnorm` density or mass fn

- `sample(1:6, 10, replace=T)`

Distribution	Functions
Beta	`pbeta`
Binomial	`pbinom`
Cauchy	`pcauchy`
Chi-Square	`pchisq`
Exponential	`pexp`
F	`pf`
Gamma	`pgamma`
Geometric	`pgeom`
Hypergeometric	`phyper`
Logistic	`plogis`
Log Normal	`plnorm`
Negative Binomial	`pnbinom`
Normal	`pnorm`
Poisson	`ppois`
Student t	`pt`
Uniform	`punif`
Weibull	`pweibull`

One-channel queueing model

```
simOneChannelQ <- function(nJob, mInter, mServ){
  interarrivalTime <- rgamma(nJob, rate=1/mInter, shape=1);
  serviceTime <- rgamma(nJob, rate=1/mServ, shape=1);

  arrivalTime <- rep(0, nJob);
  startServiceTime <- rep(0,nJob);
  endServiceTime <- rep(0, nJob);
  waitingInQueue <- rep(0, nJob);
  waitingInSystem <- rep(0, nJob);

  for (t in 1:nJob){
    if (t == 1){
        arrivalTime[1] <- interarrivalTime[1];
        startServiceTime[1] <- arrivalTime[1];
    }else{
       arrivalTime[t] <- arrivalTime[t-1] + interarrivalTime[t];
       startServiceTime[t] <- max(endServiceTime[t-1], arrivalTime[t]);
    }
       endServiceTime[t] <- arrivalTime[t] + serviceTime[t];
        waitingInQueue[t] <- startServiceTime[t] - arrivalTime[t];
        waitingInSystem[t] <- waitingInQueue[t] + serviceTime[t];
    }
    output <- data.frame(interarrivalTime, serviceTime, arrivalTime,
    startServiceTime, endServiceTime, waitingInQueue, waitingInSystem);
    return(output)
}
```

Job No.	Interarrival Time	Service Time	Arrival Time	Start Service Time	Finish Service Time	Waiting in Queue	Time in System
0			0	0	0		
1	3.59	3.35	3.59	3.59	6.94	0	3.35
2	1.82	2.83	5.41	6.94	9.77	1.53	4.36
3	2.12	0.4	7.53	9.77	10.17	2.24	2.64
4	4.67	0.38	12.2	12.2	12.58	0	0.38
5	0.56	5.16	12.76	12.76	17.92	0	5.16

Simulation *vs* Probabilistic queuing models

- Warmup period
- Little's queuing formula $L = \lambda W$
- Steady-state distribution

π_j = P(there are *j* customers in system)

= Long-run fraction of time that there are *j* customers in system

M/M/1

- Exponential interarrival time w/ rate λ (mean inter.time $1/\lambda$)
- Exponential service time w/ rate μ (mean service time = $1/\mu$)
- Traffic intensity $\rho = \frac{\lambda}{\mu}$

- Steady state distribution
  - $\pi_j = \rho^j(1-\rho); j = 0, 1, 2, \ldots$
  - In particular, $\pi_0 = 1 - \rho$
- $L = \frac{\rho}{1-\rho}$, $W = \frac{1}{\mu-\lambda}$
- $L_q = \frac{\rho^2}{1-\rho}$, $W_q = \frac{\lambda}{\mu(\mu-\lambda)}$

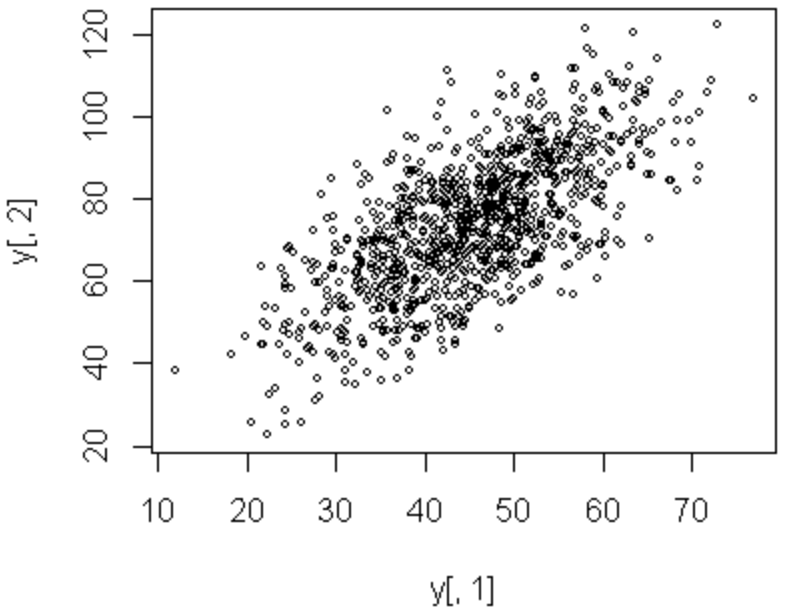
Dependent Inputs: Bivariate normal distribution

- Marginal distribution

	Mean	Stdev
X	45	10
Y	75	17

- $corr(X,Y) = \frac{cov(X,Y)}{\sqrt{var(X)var(Y)}}$
- Correlation between the two demands $\rho = 0.7$
- Covariance matrix

Covariance mx	
Var(X)	Cov(X,Y)
Cov(Y,X)	Var(Y)




```
mu1 <- 45; mu2 <- 74;
s1 <- 10; s2 <- 17; cr <- 0.7
sMx <- matrix(c(s1^2, cr*s1*s2,
                cr*s1*s2, s2^2),
                nrow=2, byrow=T)
library(MASS)
y <- mvrnorm(1000, c(mu1, mu2), sMx)
```

Correlated Inputs: Copulas

Motivation

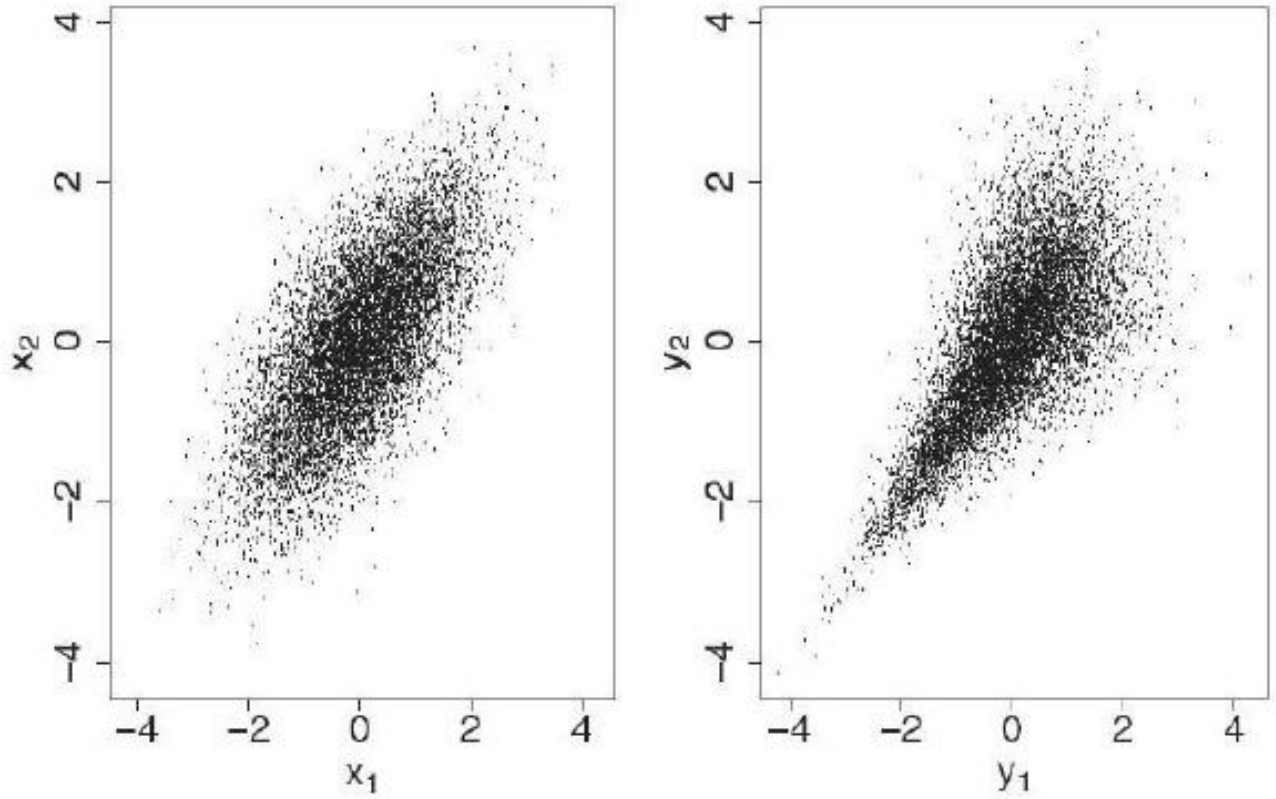


Figure: Simulations from two models, both with standard normal margins and correlation 0.7.

D. Berg. 2008. Using Copulas: An introduction for practitioners.

Sklar's theorem

Let X and Y be random variables with distribution functions F and G respectively and joint distribution function H. Then there exists a copula C such that for all $(x, y) \in \overline{\mathbb{R}} \times \overline{\mathbb{R}}$:

$$H(x, y) = C\big(F(x), G(y)\big)$$

C is unique if F and G are continuous; otherwise, C is uniquely determined on $\operatorname{ran}(F) \times \operatorname{ran}(G)$. Conversely, if C is a copula and F and G are distribution functions then the function H defined as above is a joint distribution function with margins F and G.

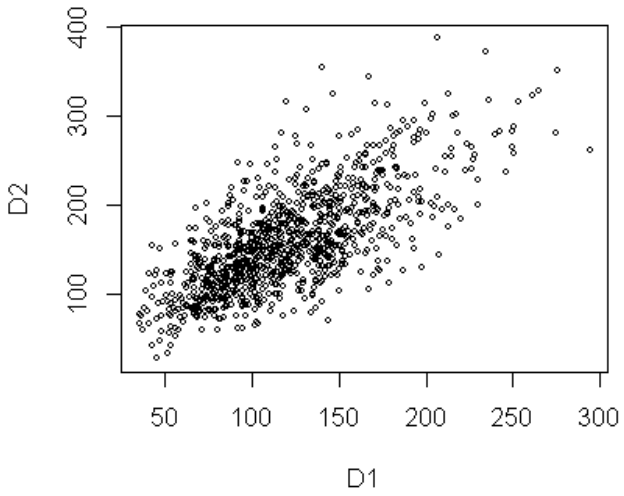
Simulating from Gaussian copula, the formula that killed Wall Street

Guassian copula $C(F, G, \mathbf{\Sigma})$

1. Generate $(X, Y) \sim N(\mathbf{0}, \mathbf{\Sigma})$
2. Take

$$F^{-1}\big(\Phi(X)\big), G^{-1}\big(\Phi(Y)\big)$$

```
> n <- 1000
> shape1 <- 8; scale1 <- 15;
> shape2 <- 8; scale2 <- 20;
> x <- mvrnorm(n, c(0,0),
                matrix(c(1,0.7,0.7,1), nrow=2,byrow=T) )
> v <- pnorm(x)
> D1 <- qgamma(v[,1], shape=shape1, scale=scale1)
> D2 <- qgamma(v[,2], shape=shape2, scale=scale2)
> plot(D1, D2, cex=0.5)
> cor(D1,D2)
[1] 0.7162509
```



Contact information

รศ. ดร. กาญจ์นภา อมรัชกุล

- อาจารย์ประจำหลักสูตรการจัดการโลจิสติกส์
- kamaruchkul@as.nida.ac.th

- การศึกษา
  - Ph.D., Industrial Engineering, Minor Statistics, University of Minnesota, Twin Cities, MN, USA, 2007
  - M.S., Industrial Engineering and Operations Research, University of California, Berkeley, CA, USA, 2003
  - A.B., Mathematics (Honors), Princeton University, Princeton, NJ, USA 2001
- ความเชี่ยวชาญ
  - Revenue management
  - Supply chain and logistics management
  - Stochastic models
  - Applied operations research

หลักสูตรและสาขาวิชาทั้งหมดของคณะสถิติประยุกต์

- วิทยาการคอมพิวเตอร์และระบบสารสนเทศ (Computer and Information Systems)
  - วิทยาการคอมพิวเตอร์ (Computer Science)
  - การจัดการระบบสารสนเทศ (Information Systems Management)
  - สถาปัตยกรรมซอฟต์แวร์ (Software Architecture)
  - วิทยาการข้อมูล (Data Science)
  - ความมั่นคงสารสนเทศ (Information Security)
  - วิทยาการสื่อปฏิสัมพันธ์ (InterActive Media Science)
- สถิติประยุกต์ (Applied Statistics)
  - สถิติ (Statistics)
  - วิทยาการประกันภัยและการบริหารความเสี่ยง (Insurance Actuarial Science & Risk Management)
  - ปัญญาและวิเคราะห์ธุรกิจ (Business Analytics & Intelligence)
- การจัดการโลจิสติกส์ (Logistics Management)
- การบริหารเทคโนโลยีสารสนเทศ (Information Technology Management)

Website
http://www.as.nida.ac.th/
http://www.logisticsatnida.net/

Facebook
School of Applied Statistics, NIDA
Logistics Management at NIDA