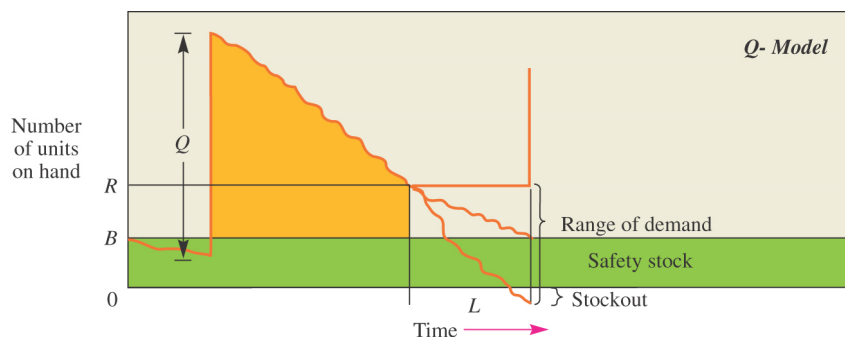


## Chapter 4 Terminating Simulation

- EXCEL庫存管理模擬
- ARENA庫存管理模擬
- What is Terminating Simulation?
- Project Management

### 4.1 固定補貨量模式 Q model

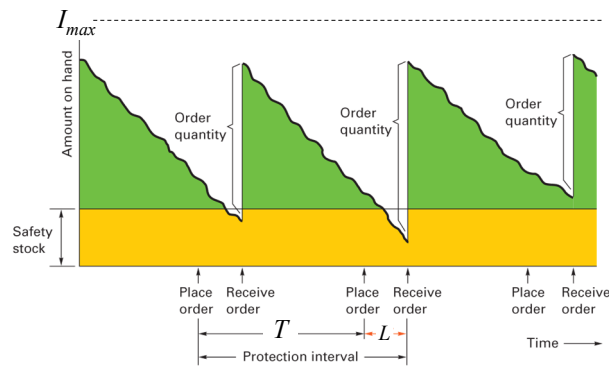
隨時監控庫存，當庫存量加上已訂購量降至訂貨點 $R$ ，就發出固定數量 $Q$ 的訂單，補貨需要一段時間 $L$ 後才到達。



數學公式須假設需求沒有尖離峰起伏

## 固定補貨週期模式 P model

每隔固定的時間 $T$ 檢查庫存，然後發出訂單將庫存量補足到庫存上限 $I_{max}$ ，但是補貨需要一段時間 $L$ 後才到達。



數學公式須假設需求沒有尖離峰起伏

## 固定週期管理的範例與公式

$T = 1$  week, 每週(第 $t$ 週)結束時進行庫存管理

$L = 2$  weeks, 兩週後(第 $t+3$ 週開始時)補貨可供使用

$B(t)$ : 第 $t$ 週的期初庫存     $D(t)$ : 第 $t$ 週的需求 (隨機發生)

$S(t)$ : 第 $t$ 週的銷售量     $E(t)$ : 第 $t$ 週的期末庫存

$O(t)$ : 第 $t$ 週的訂貨量

- $B(t) = E(t-1) + O(t-3)$  期初庫存=上週期末庫存+兩週前的訂單量
- $S(t) = \min(B(t), D(t))$  銷售量不大於期初庫存或需求量
- $E(t) = B(t) - S(t)$  期末庫存=期初庫存 - 銷售量
- $O(t) = \max(0, I_{max} - E(t) - O(t-1) - O(t-2))$   
期末庫存 已訂而尚未送達的數量

## 試算表模擬庫存控管

Inventory.xlsx

H5     fx     =MAX(H\$2-(E5+F5+G5),0)

B	C	D	E	F	G	H	I	J	K	L	M	N
						庫存上限	單位利潤	單位持有	單位缺貨	檢查成本	訂貨成本	
						400	\$5	\$1	\$2	\$30	\$70	
期初庫存	顧客需求	銷售量	期末庫存	在途 1	在途 2	訂貨量	銷售利潤	持有成本	缺貨損失	檢查成本	訂貨成本	盈餘
400	213	213	187	0	0	213	\$1,065	\$293.50	\$0	\$30	\$70	\$672
187	179	179	8	213	0	179	\$895	\$310.50	\$0	\$30	\$70	\$485
8	182	8	0	179	213	8	\$40	\$396.00	\$348	\$30	\$70	(\$804)
213	177	177	36	8	179	177	\$885	\$311.50	\$0	\$30	\$70	\$474
215	282	215	0	177	8	215	\$1,075	\$292.50	\$134	\$30	\$70	\$549
8	218	8	0	215	177	8	\$40	\$396.00	\$420	\$30	\$70	(\$876)
177	255	177	0	8	215	177	\$885	\$311.50	\$156	\$30	\$70	\$318
146	267	146	0	134	120	146	\$730	\$327.00	\$242	\$30	\$70	\$61
120	201	120	0	146	134	120	\$600	\$340.00	\$162	\$30	\$70	(\$2)
134	176	134	0	120	146	134	\$670	\$333.00	\$84	\$30	\$70	\$153
146	161	146	0	134	120	146	\$730	\$327.00	\$30	\$30	\$70	\$273
120	128	120	0	146	134	120	\$600	\$340.00	\$16	\$30	\$70	\$144
148.46	183.23	129.48	18.98				\$647.40	\$335.26	\$107.50			\$104.6

Exercise: 請以10為單位調整庫存上限，使每週的平均盈餘超過\$200。

## 4.2 選擇補貨系統

### ● Option Replenishment System

綜合P model與Q model特性的補貨模式，管理者定期檢查庫存，當檢查時發現庫存量加已訂購量降至訂貨點以下，將發出訂單以補足庫存，否則留待下期檢查再做決定。

- 可控制參數：檢查週期、訂貨點、庫存上限。
- 影響的系統狀態：補貨次數、庫存量、缺貨數量
- 相關成本為：檢查成本、補貨的作業成本、缺貨成本、庫存持有成本。不計銷售利潤。
- 目標：以最低成本滿足大多數的需求

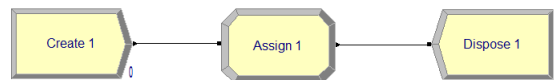
## 初步的模擬設計

- 兩個平行的流程，一個模擬隨時都會發生的內部需求，另一個模擬依固定週期檢查與補充庫存的工作。
- 兩種entity，內部顧客在第一個流程裡提出需求，庫存管理工作在第二個流程裡盤點並補充庫存。
- Attribute是entity的共同屬性，內容是各個entity的資料。  
demand為顧客的需求量，order size為庫存盤點後的訂單量
- Variable描述系統狀態，任何entity在同一時間會讀取到相同數值。  
inventory是現有庫存量，total usage是累計使用量，total shortage是累計缺貨量，order count是累計補貨次數

**warning:** 要設定的資訊是attribute? 還是variable?

## 內部需求的流程模組

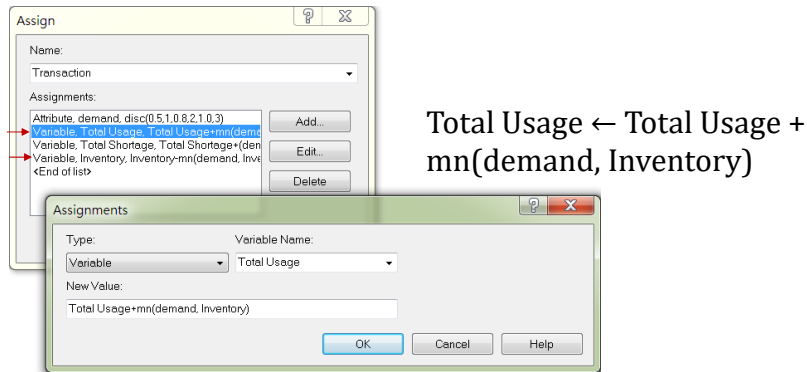
- 內部的需求隨機到達，平均一小時發生一次。



- **Assign**模組可設定attribute, variable，記錄或修改數值
- 確定需求量、累計使用量與缺貨量、更新庫存量

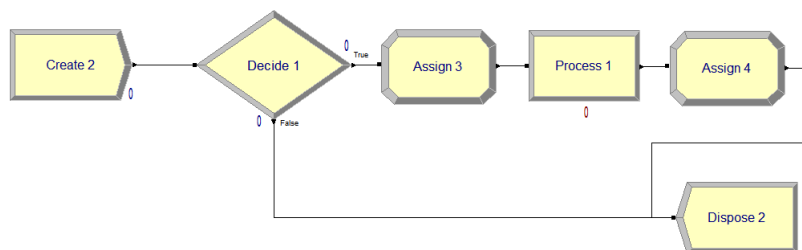
缺貨時改向外緊急採購，不等補貨，只需計算缺貨成本

## Assign模組紀錄與更新數值



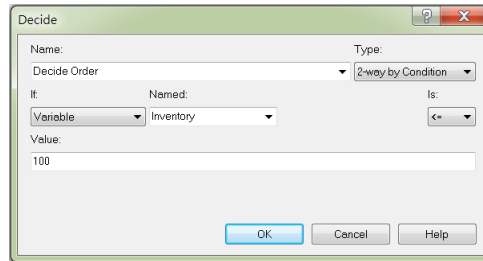
- Type: Attribute or Variable
- demand=disc(0.5,1, 0.8,2, 1.0,3)
- 最後執行：Inventory ← Inventory – mn(demand, Inventory)

## 庫存管理者的流程



- Create模組：到達時間間隔改為Constant，數值為7 days
- Decide模組：依照系統狀態決定是否訂貨
- Assign模組：分別更新訂貨與庫存紀錄
- Process模組：模擬交貨所需時間

## Decide模組根據邏輯條件分流

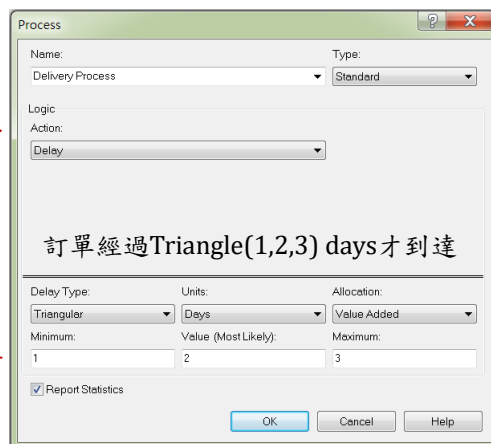
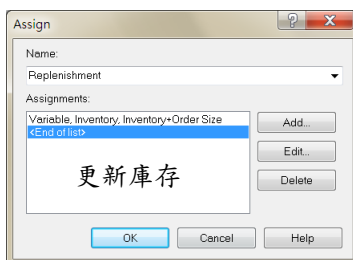
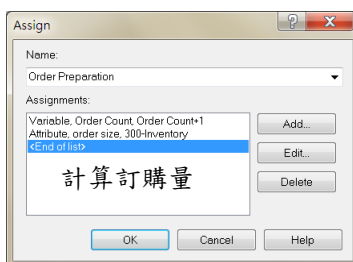


- Type: 2-way by Condition
- 庫存降至訂貨點或更低：Inventory <= 100
- 條件成立，選擇True出口，否則選擇False出口



管理週期大於交貨時間，決策時不需考量在途訂單

## 訂單補貨過程



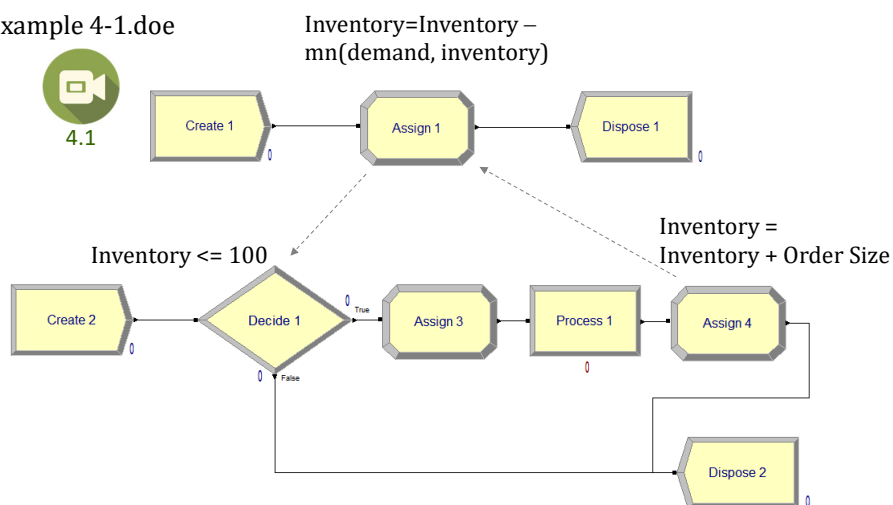
## Attributes vs. Variables

Variable - Basic Process										
	Name	Comment	Rows	Columns	Data Type	Clear Option	File Name	Initial Values	Report Statistics	Initial Values
1	Inventory				Real	System		1 rows	<input checked="" type="checkbox"/>	
2	Order Count				Real	System		0 rows	<input type="checkbox"/>	
3	Total Usage				Real	System		0 rows	<input type="checkbox"/>	100
4	Total Shortage				Real	System		0 rows	<input type="checkbox"/>	

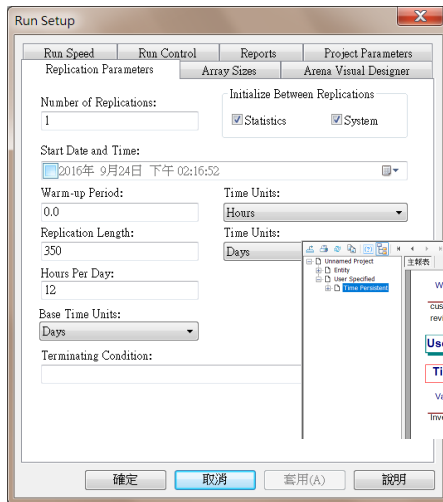
- **Attribute**是**entity**的共同屬性，內容是各個**entity**的資料。好比每張身分證都有相同欄位，但是欄位的記載內容不同。例如 **demand**是每個顧客的需求量，或是不同訂單的訂購內容。
- **Variable**是記錄整個系統的狀態，同一時刻，任何**entity**都會讀取到相同數值。例如航班售票狀況或是某種商品庫存量。
- 將**inventory**誤設為**attribute**，可能使不同顧客讀取到不同的庫存量，好比乘客上網看到完全不同的售票狀況。
- 將**order size**誤設為**variable**，會使先前的訂單量被後續的訂單更改。好比你網購的內容被下一個下單的顧客更改。

## Variables: 兩個流程的隱形連結

Example 4-1.doe



## 正確但無助益的模擬結果



Example 4-1.doe

	Total time	WIP
Customer	0	0
Reviewer	0.9786	0.1398

Variable	Average	Half Width	Minimum Value	Maximum Value
customer	0.00	(insufficient)	0.00	1.0000
reviewer	0.1398	(insufficient)	0.00	1.0000

Variable	Average	Half Width	Minimum Value	Maximum Value
Inventory	133.14	5.33179	0.00	300.00

Q: 模擬之目的為何?

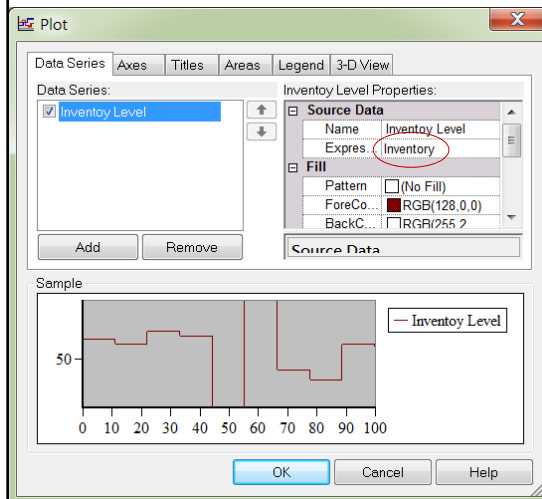
## 4.3 建立能支援決策的模擬模式

為了解庫存變化並建立改善的依據：

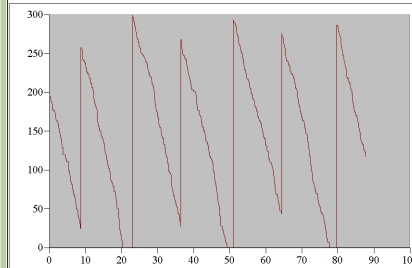
- 將檢查週期(period)、訂貨點(reorder point)、庫存上限(limit)設為variables。
- 加入動畫顯示庫存量變化
- 使用Statistics模組定義各種績效值與成本。



## 折線圖顯示庫存量的變化

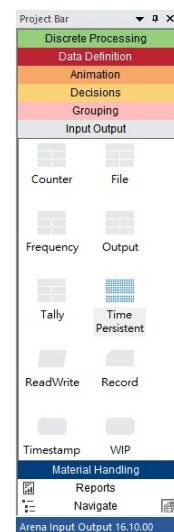


Line > DrawMode > Stairs



## 常見的績效類型與計算方式

- **Time-persistent** variable: 隨時間而變的數值，如庫存或排隊人數 (discrete change variable)。溫度?
- 常見的績效是數值在一段時間內的平均值，如二月份平均庫存，稱為time-based average。
- 計算平均值時是計算總面積，再除以觀察的時間長度。令 $N(t)$ 為時間 $t$ 時的排隊人數，則 $[0, T]$ 之間的平均值可以表成
 
$$\frac{\int_0^T N(t)dt}{T}$$
- 離散變數呈現階梯式的變化，很容易計算面積與平均高度



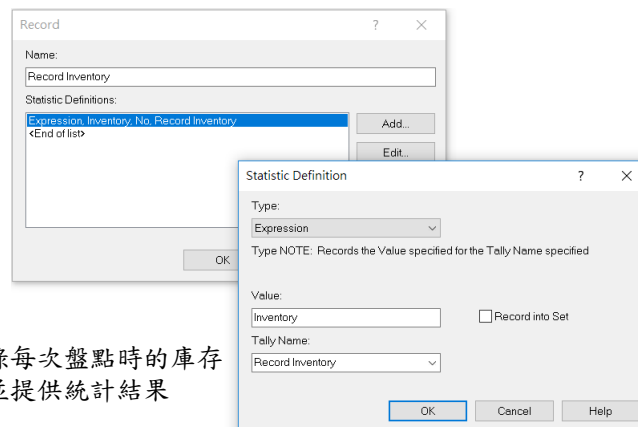
## 常見的績效類型與計算方式

- **Tally variable:** 同類型的觀察值，如各月份的營業額、每個週期盤點的庫存量。
- 計算平均值是將觀察值的總和除以觀察數，如先後到達顧客之平均等候時間，又稱為 **observation-based average**。設  $x_1, x_2, \dots, x_n$  是同類型的數值，平均值是：

$$\frac{\sum_{i=1}^n x_i}{n}$$

- **Counter:** 計數器，累計事件發生的次數，如商品缺貨的次數，也可以用來計算事件發生的比例，如缺貨率。

## Record模組記錄Tally變數



記錄每次盤點時的庫存量並提供統計結果

## Statistic模組與Time Persistent變數

Variable - Basic Process										
	Name	Comment	Rows	Columns	Data Type	Clear Option	File Name	Initial Values	Report Statistics	Initial Values
1	Inventory				Real	System		1 rows	<input checked="" type="checkbox"/>	
2	Order Count				Real	System		0 rows	<input type="checkbox"/>	
3	Total Usage				Real	System		0 rows	<input type="checkbox"/>	100
4	Total Shortage				Real	System		0 rows	<input type="checkbox"/>	

Type: Time-Persistent

Arena全程追蹤庫存變化並計算平均庫存量

Statistic - Advanced Process						
	Name	Type	Expression	Collection Period	Report Label	Output File
1	inventory cost	Time-Persistent	Inventory*0.02	Entire Replication	inventory cost	

Double-click here to add a new row.

Expression: Inventory\*0.02

Arena會追蹤庫存成本變化並計算平均庫存成本  
=DAVG(inventory cost)

## Output設定在模擬結束時計算績效值

Statistic - Advanced Process					
	Name	Type	Expression	Collection Period	Report Label
1	inventory cost	Time-Persistent	Inventory*0.02	Entire Replication	inventory cost
2	daily cost	Output	(60*EntitiesIn(reviewer)+200*order count+5*total shortage)/350 +DAVG(inventory cost)	Entire Replication	daily cost
3	total usage count	Output	Total Usage	Entire Replication	total usage count
4	total shortage count	Output	Total Shortage	Entire Replication	total shortage count
5	total order count	Output	Order count	Entire Replication	total order count

Double-click here to add a new row.

Expression Builder

Expression Type: Basic Process Variables

Entity Type: reviewer

Current Expression:  
(60\*EntitiesIn(reviewer)+200\*order count+5\*total shortage)/350 +DAVG(inventory cost)

Type: Output

單日全部成本=檢查成本+補貨成本+缺貨成本+持有成本

$$= (60 * \text{EntitiesIn}(\text{reviewer}) + 200 * \text{order count} + 5 * \text{total shortage}) / 350 + \text{DAVG}(\text{inventory cost})$$

## 單次350天的模擬結果

Example 4-2.doe

The screenshot displays the following data:

User Specified				
Tally				
Expression	Average	Half Width	Minimum Value	Maximum Value
Record Inventory	102.82	(Insufficient)	0.00	221.00
Time Persistent				
Time Persistent	Average	Half Width	Minimum Value	Maximum Value
inventory cost	2.6978	0.109258719	0.00	6.0000
Output				
Output	Value			
daily cost	32.2692			
total order count	25.0000			
total shortage count	470.00			
total usage count	6795.00			

## 4.4 模擬的運作型態

**Terminating simulation**：有既定的起始與結束的條件，但是運作時間長度未必固定，也不一定是週期性的運作。

- 捷運從清晨六點到半夜十二點的營運週期。
- 專案或戰鬥模擬有既定的結束條件，但沒有既定的時刻。

**Steady state simulation**：連續長時間運作，無週期或結束條件

- 生產線即使夜晚停工，在製品仍然留在線上，第二天早上開工時，從昨天下班時的狀態繼續運作。
- 港口的船舶進出管制

兩種模擬型態需要不同設定與不同的分析方式。

## Schedule模組設定尖離峰的需求

設需求以年為周期，有尖離峰變化，則可視為Terminating Simulation

時間 (days)	1-50	51-100	101-150	151-200	201-250	251-300	301-350
每小時到達數 (arrivals/hour)	0.9	1.0	1.05	1.1	1.05	1.0	0.9

Schedule資料模組 Type: Arrival

1.1/hour for 50 days →

	Value	Duration
1	0.9	50
2	1	50
3	1.05	50
4	1.1	50
5	1.05	50
6	1	50
7	0.9	50

## Create模組的需求到達間隔

- 尖離峰的需求到達間隔平均值配合 Demand Schedule的設定而調整

1-50: 平均到達間隔=1/0.9 hour

50-100: 平均到達間隔=1 hour

100-150: 平均到達間隔=1/1.05

## 尖離峰變化對績效的影響

Example 4-3.doe (1 replication)

Expression	Average	Half Width	Minimum Value	Maximum Value
Record Inventory	99.94	(Insufficient)	0.00	211.00

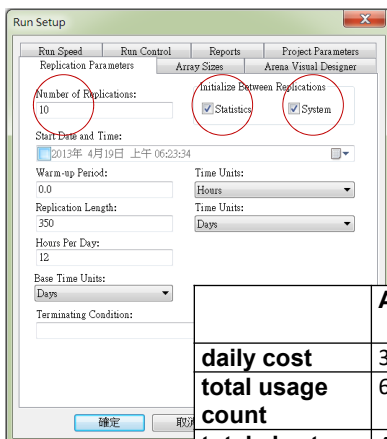
  

Time Persistent	Average	Half Width	Minimum Value	Maximum Value
inventory cost	2.6564	0.148211145	0.00	6.0000

Output	Value
daily cost	33.8564
total lost sales count	584.00
total order count	25.0000
total sales count	6860.00

## 4.5 可信賴區間與績效比較



Terminating simulation

⇒ IID observations

⇒ 95% confidence intervals

無尖離峰變化，10次模擬結果的平均值、半寬、最小與最大值

	Average	Half-width	Min. Average	Max. Average
daily cost	31.6612	0.72	29.8163	32.6067
total usage count	6695.3	50.56	6596	6795
total shortage count	428.80	49.90	299	496

32.2692  
470

## 降低可信賴區間半寬

$$n \text{ replications} \quad h_1 = t_{n-1, \alpha/2} \frac{S_n}{\sqrt{n}} \quad 2n \text{ replications} \quad h_2 = t_{2n-1, \alpha/2} \frac{S_{2n}}{\sqrt{2n}}$$

$$\frac{h_2}{h_1} = \frac{1}{\sqrt{2}} \approx 0.707$$

希望將半寬由  $h_1$  降至  $h \Rightarrow$  所需的 replications  $\cong n \left( \frac{h_1}{h} \right)^2$

## 降低模擬誤差以獲得有效結論

20 replications

Example 4-2.doe 無尖離峰變化

	Average	Half-width
daily cost	31.8001	0.43
total usage count	6686.75	37.15
total shortage count	438.40	30.56

32.2692  
470

Example 4-3.doe 有尖離峰變化

	Average	Half-width
daily cost	32.4286	0.50
total usage count	6672.45	43.77
total shortage count	481.10	35.36

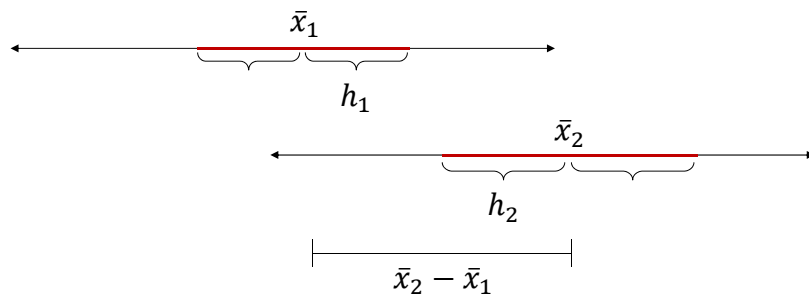
100 replications

無尖離峰變化	Average	Half-width	有尖離峰需求	Average	Half-width
daily cost	31.9454	0.21	daily cost	32.4341	0.22
total usage count	6687.66	15.06	total usage count	6677.56	18.10
total shortage count	450.04	14.88	total shortage count	482.07	15.73

Caution: 使用者應該思考是否有流程差異或合適理由足以解釋績效的差距。有可能沒有差距或是差距小至沒有實質意義。

## 簡易判斷兩者差距是否顯著

ARENA分別模擬兩個系統並計算可信賴區間

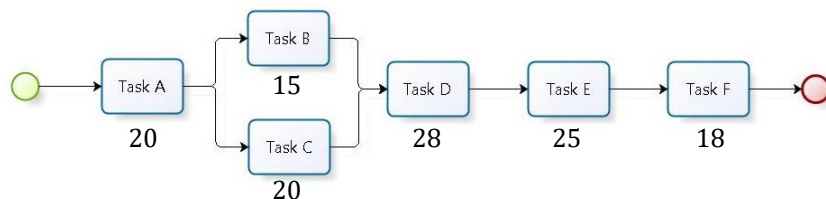


樣本平均值的差距  $\bar{D} = \bar{x}_2 - \bar{x}_1$  大於半寬的和  $h_1 + h_2 \Rightarrow$  顯著差異

## 4.6 Project Management

專案是由一群具有執行順序與共同目標的工作所組成，各項作業所需時間無法確知，經常表示為機率分佈

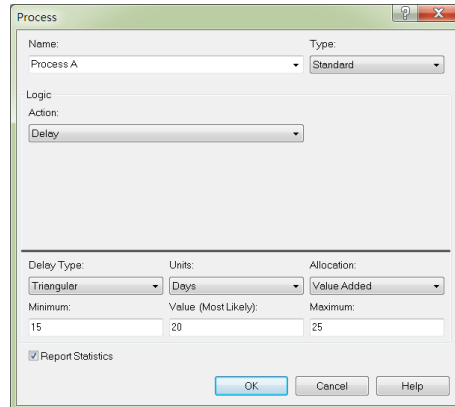
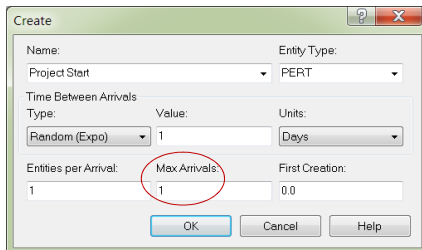
作業	時間(天數)	先行作業
A	tria(15,20,25)	—
B	tria(10,15,25)	A
C	unif(10,30)	A
D	tria(22,28,36)	B, C
E	tria(20,25,30)	D
F	tria(12,18,24)	E





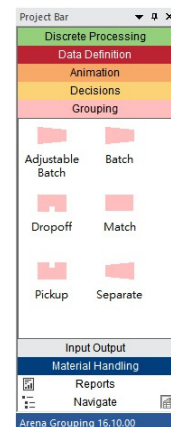
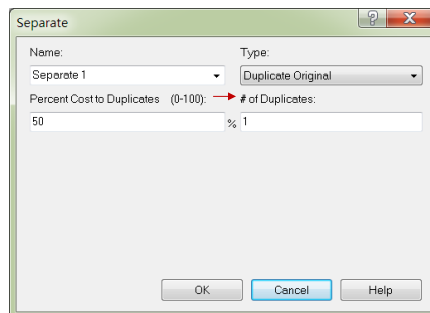
## Create與Process模組

- Create模組代表專案開始
- Max Arrivals欄位改設為1
- Time Between Arrivals的設定不會有任何作用
- Process模組代表各項作業過程



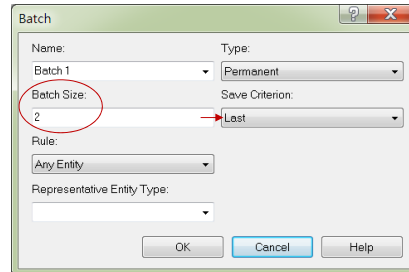
## Separate模組將流程一分為二

- 使用模組預設值
- 原先進入的entity從標示**Original**的出口離開，複製的分身從標示**Duplicate**的出口離開
- 本尊與分身分別連到代表B與C的Process模組

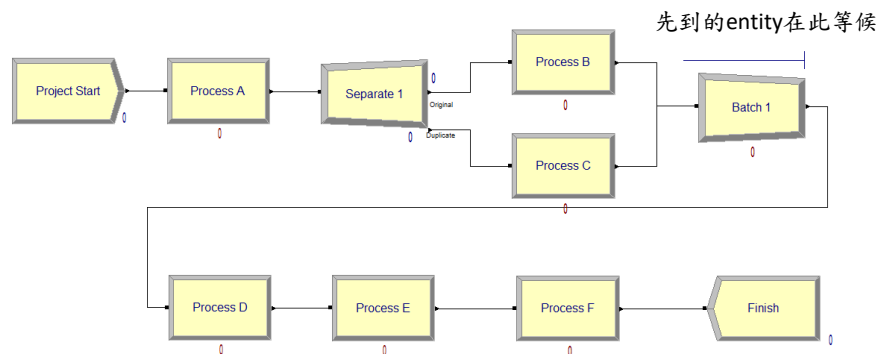


## Batch模組匯集不同的流程

- 全部使用模組預設值
- Batch Size: 2  
須累計兩個entity進入模組，才滿足合併並離開的條件
- Save Criterion: Last  
不論本尊或分身，較晚進入者成為合併後的entity

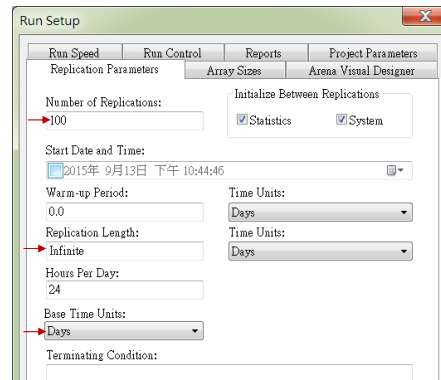


## 專案的全部流程



## 模擬實驗設定

- 無法預知專案何時結束
- Replication Length維持預設值Infinite
- 當專案完成並離開流程時，因為無未來將發生的事件，Arena會自動結束模擬
- 以不同亂數進行100次模擬



Example 4-4.doe

Total Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
PERT	113.33	1.52	89.3115	133.14	89.3115	133.14
Other						

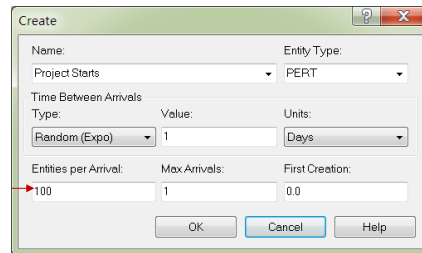
## 4.7 不確定的專案架構

- 作業D完成後進行評估，有10%的機率會停止並放棄專案，另有30%的機率要進行額外的修正作業
- 作業E完成後，如果專案執行已超過100天，則增加資源以加速完成作業F。
- 以110天為期限，估算專案如期完成的機率以及超過期限時的平均延誤天數。

## 更改模擬進行方式

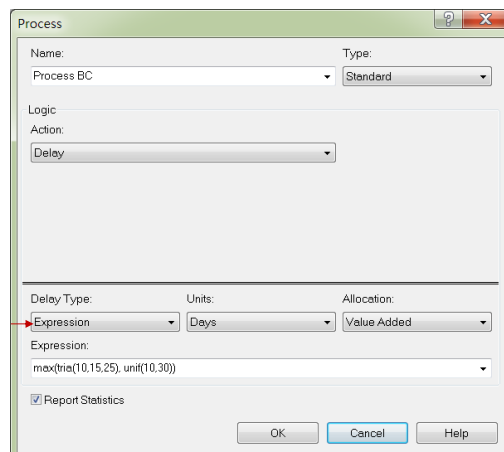
先前做法：每次模擬一個專案，  
以不同亂數執行100 replications

新方式：Entities per Arrival=100  
模擬開始就一次產生100個專案，  
各自進行，不互相影響



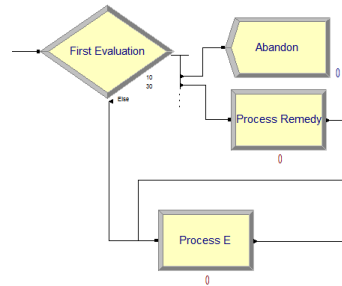
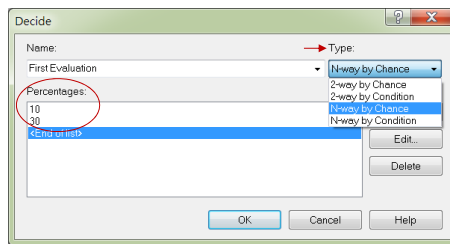
## 模擬作業B, C的進行

- Delay Type的設定改為 Expression
- $\max(\text{tria}(10,15,25), \text{unif}(10,30))$ ，代表完成B與C所需的時間
- 取代原先的Process B, Process C, Separate, Batch模組



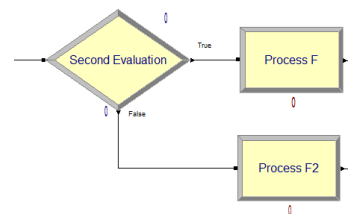
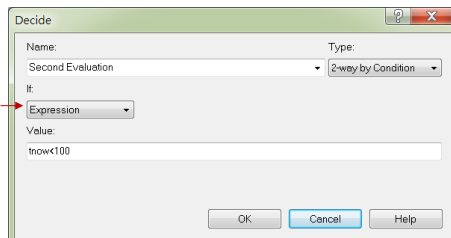
## Decide模組隨機選擇可能結果

- 評估是否放棄或需要修正專案
- N-way by Chance: Arena使用亂數來選擇出口
- 10%進入Dispose模組，取消勾選Record Entity Statistics
- 30%修正所需的的天數為 $\text{expo}(10)$
- 其餘從底部的Else出口離開，不需修正



## Decide模組以邏輯條件判斷

- 決定是否對作業F進行趕工
- 2-way by Condition:  $\text{tnow} < 100$  (days)
- $\text{tnow}$ 是Arena內建變數，數值為讀取時的模擬時刻
- 條件不成立，從False出口離開，趕工天數為 $\text{unif}(6,14)$



## Record模組紀錄績效

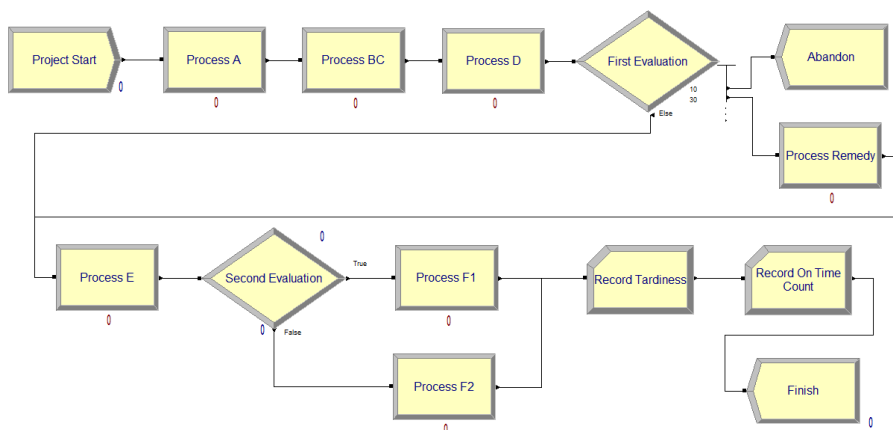
- $\max(\text{tnow}-110,0)$  計算延誤天數
- Type: Expression 計算平均值、最小與最大值 (不含放棄的專案)

The screenshot shows a 'Record' dialog box with the following fields:  
Name: Record Tardiness  
Type: Expression  
Value:  $\max(\text{tnow}-110,0)$   
Record into Set:   
Tally Name: project tardiness  
Buttons: OK, Cancel, Help

The screenshot shows a 'Record' dialog box with the following fields:  
Name: Record On Time Count  
Type: Count  
Value:  $\min(\text{tnow}-110,0)/(\text{tnow}-110)$   
Record into Set:   
Counter Name: Record On Time Count  
Buttons: OK, Cancel, Help

- $\min(\text{tnow}-110,0)/(\text{tnow}-110)=1$  代表專案未延誤
- 可改用邏輯判斷式  $\text{tnow} < 110$
- Type: Count 自動累計未延誤的專案次數
- 第15版可合併成一個Record模組

## 修改後的專案流程



## 單次模擬結果

Example 4-5.doe

Replications: 1 Time Units: Days

**PERT**

**Key Performance Indicators**

**System**  
Number Out Average 88

Expression	Average	Half Width	Minimum Value	Maximum Value
PERT	113.94	(Insufficient)	95.0393	163.22

**Other**

Number In Value

PERT	100.00
------	--------

**User Specified**

**Tally**

Expression	Average	Half Width	Minimum Value	Maximum Value
project tardiness	5.1506	(Insufficient)	0.00	53.2198

**Counter**

Count Value

Record On Time Count	23.0000
----------------------	---------

NKFUST Simulation Project Management Slide 45

## 估算如期完工的機率

- 100次專案只能產生一個如期完成的比例
- Number of Replications=10，模擬後就能得到10個機率值
- Arena計算平均值與可信賴區間半寬

### Tally

Expression	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
project tardiness	4.4999	0.60	3.3541	5.7867	0.00	53.938

### Counter

Count	Average	Half Width	Minimum Average	Maximum Average
Record On Time Count	25.9000	3.22	18.0000	32.0000

專案如期完工的機率約為 $25.9 \pm 3.22\%$   
(包含放棄的專案)