PROGRAMACIÓN ESTOCÁSTICA: MODELO DETERMINISTA EQUIVALENTE Y SU IMPLEMENTACIÓN EN EL ENTORNO DEL SOFTWARE R

Mg. Fabián Andrés Ferreira
Doctorado en Ingeniería de la Decisión
Universidad Rey Juan Carlos

Dr. Andrés Redchuk
Facultad de Ingeniería
Universidad Nacional de Lomas de Zamora. Argentina

RESUMEN

En este trabajo se desarrolla e implementa en el software R un algoritmo específico para resolver ciertos sistemas lineales estocásticos. Se utilizan desigualdades probabilísticas y se obtiene la solución del problema determinista equivalente resultante. Para la resolución de los problemas deterministas se utiliza el lpSolver. Se automatiza todo el proceso, y se brinda un amplio soporte gráfico.

Palabras Clave: Programación Estocástica, Software R, lpSolver.

ABSTRACT

In this work we have developed and implemented in the software R a specific algorithm for solving certain linear stochastic systems. We obtain the solution of the equivalent deterministic resulting problem by using probabilistic inequalities. To solve the deterministic problems we have used the lpSolver. The process is completely automatised, and we provide an extensive graphic support.

Keywords: Stochastic Programming, Software R, lpSolver.

1. INTRODUCCIÓN

En el campo de la Optimización bajo incertidumbre, las distintas técnicas de programación estocástica; buscan solucionar problemas de asignación de recursos, donde uno o varios parámetros son desconocidos en el momento de tomar la decisión; pero se conoce o se puede estimar por ejemplo, a partir de observaciones previas; su distribución probabilística [2].

Normalmente las distintas técnicas empleadas, buscan en última instancia reducir el problema estocástico a un problema determinista; que se llama determinista equivalente, cuya solución óptima pasa a considerarse la solución optima del problema estocástico.

Nos centraremos en los modelos de programación estocástica activa o “aquí y ahora” (“here and now”), en los cuales el decisor toma la decisión sin el conocimiento de la realización de las variables aleatorias; solo contando con o pudiendo estimar las distribuciones de probabilidad de las mismas [9]. Dentro de este tipo de modelos se enmarca el problema planteado, existiendo diferentes enfoques para abordar este tipo de problemas:

Los *modelos estocásticos de programación*, por contra; incorporan dentro del modelo las distribuciones de probabilidad que describen los datos, bien porque se conozcan o por que se puedan estimar. Siendo el objetivo, encontrar alguna solución factible para todos (o casi todas) las instancias de datos posibles; consiguiendo simultáneamente maximizar el valor esperado de la función objetivo [9].

Una técnica muy empleada dentro de la programación estocástica activa, es la *aproximación por escenarios* [6] para representar el posible desempeño futuro de las decisiones adoptadas; con lo que se hace posible calcular una solución al problema de programación estocástica, mediante la solución de un problema de programación determinista equivalente para cada escenario; esto es posible cuando se cuenta con una distribución de probabilidad discreta de la variable aleatoria, y por tanto el análisis puede centrarse en una cantidad finita (pequeña) de escenarios.

Cuando las distribuciones de probabilidad de los parámetros aleatorios son continuas, o existen muchos parámetros aleatorios; se dificulta la construcción de escenarios apropiados para aproximar la incertidumbre. Una solución habitual consiste en construir dos modelos deterministas equivalentes distintos, para las soluciones óptimas que proporcionan los límites superior e inferior para el valor óptimo (entendido este como la esperanza matemática de la variable aleatoria).

Aproximaciones alternativas para obtener soluciones válidas a este tipo de problemas incluyen:

El *método de restricciones aleatorias* [2] consiste en transformar el problema dado en un determinista equivalente, en el que se verifiquen las restricciones con al menos una determinada probabilidad fijada de antemano; que refleje el nivel de riesgo asumible por parte del decisor, de que la solución del problema sea no factible. Hay que distinguir dos casos, según se fije la probabilidad para el conjunto de las restricciones; o para cada una de ellas por separado.

En la *programación lineal por etapas* se elige una solución optima del modelo en la primera etapa [13], cuyo efecto se ve modificado posteriormente por la ocurrencia de un evento aleatorio. En la segunda (y las etapas siguientes), se toma una decisión que busca corregir las desviaciones causadas por dicho evento aleatorio; sobre la decisión tomada en la etapa precedente. La solución óptima del modelo así planteado, es una única decisión en la primera etapa; y un conjunto de acciones de recurso en la segunda etapa y siguientes, para cada resultado aleatorio que se desvíe del resultado óptimo. Todo ello implica conocer los costes de las acciones correctivas, los cuales deben incorporarse dentro de la función objetivo.

2. MÉTODOS

Para abordar el problema que nos ocupa, definimos un modelo de programación lineal estocástica; donde solo el lado derecho correspondiente a la cantidad de recursos disponibles (los Bi's), tiene un componente estocástico con una distribución de probabilidad conocida; o que en su defecto puede estimarse. Aplicaremos el método de restricciones aleatorias con distribuciones de probabilidad independientes.

Partiendo de la definición general de un problema de programación lineal determinista [1]:

$$\begin{aligned} &min \quad c^T x \\ &s.a \;\; Ax \geq b \\ &l \leq x \leq u \end{aligned} \tag{1}$$

Donde tanto el gradiente de la función objetivo C^T, como la matriz tecnológica A, y el vector del lado derecho correspondiente a los coeficientes de recursos disponibles b; son todos parámetros conocidos. Habitualmente también se supone la no negatividad de las variables, con lo que $l_i = 0$ y $u_i = +\infty$.

En nuestro problema de estudio, el vector b, correspondiente a las cantidades de recursos disponibles, no se conoce de antemano; con lo que lo hemos reemplazado con un vector estocástico ξ, lo que si conocemos (o podemos estimar) es la función de distribución de probabilidad del mismo F_ξ.

Tenemos entonces el modelo de programación lineal estocástica [9]:

$$\begin{aligned} &min \quad c^T x \\ &s.a \;\; Ax \geq \xi \\ &x \geq 0 \end{aligned} \tag{2}$$

Aquí las restricciones de la forma $Ax \geq \xi$ se pueden cumplir solo con cierto grado de probabilidad α; con lo que tendremos que reemplazarlas mediante desigualdades probabilísticas del tipo:

$$P(Ax \geq \xi) \geq \alpha \tag{3}$$

Si definimos la función compuesta G(x), talque sea igual a la función de distribución de probabilidad del vector estocástico para cada conjunto de valores Ax [5]:

$$G(x) = P_{\xi}(x \mid Ax \geq \xi) \qquad (4)$$

O lo que es lo mismo:

$$G(x) = F_{\xi}(Ax) \qquad (5)$$

Tenemos que, aun siendo un vector estocástico ξ, si conocemos F_{ξ}; con lo cual podemos transformar dicha restricción en:

$$G(x) \geq \alpha \qquad (6)$$

Que aunque no es una restricción lineal, sabiendo por la definición de la función cuantíl que:

$$P_{\xi}(x \mid Ax \geq \xi) \geq \alpha \Leftrightarrow F_{\xi}(Ax) \geq \alpha \Leftrightarrow Ax \geq Q_{\xi}^{-}(\alpha) \qquad (7)$$

Donde Q_{ξ} ( ) es la función cuartíl de ξ, y Q_{ξ}^{-} (α) es la cota mas a la izquierda del intervalo cerrado de los α–cuartiles de F_{ξ}.

En nuestro caso, partiendo de la ecuación (6) en su forma más general $G(x) \leq \beta$:

Podemos conseguir su equivalente lineal mediante [5]:

$$P_\xi(x \mid Ax \geq \xi) \leq \beta \Leftrightarrow F_\xi(Ax) \leq \beta \Leftrightarrow Ax \leq Q_\xi^+(\beta) \qquad (8)$$

Donde Q_ξ^+(β) es la cota mas a la derecha del intervalo cerrado de los b–cuartiles de F_ξ.

Las mismas son ya restricciones lineales, que nos permiten resolver el problema lineal determinista equivalente; mediante técnicas tradicionales como el método Simplex. Siempre con cierto grado de probabilidad α (que suele ser en la práctica del 95%):

$$\begin{aligned} &min \quad c^T x \\ &s.a \;\; Ax \geq Q_\xi^-(\alpha) \qquad (9) \\ &x \geq 0 \end{aligned}$$

Por otro lado, si trabajáramos con restricciones estocásticas con probabilidad conjunta, buscando que estas se cumplan con una determinada probabilidad α; las mismas se reemplazan en el modelo determinista por la restricción probabilística

$$P(a_i x \leq b_i, ..., a_n x \leq b_n) \geq \alpha \qquad (10)$$

Que en caso de ser mutuamente independientes (dichas variables estocásticas), se transforma en:

$$\prod_{i=1}^{n} P(a_i x \leq b_i) \geq \alpha \qquad (11)$$

En el caso de que se busque que cada restricción de verifique con una probabilidad distinta α_i, el riesgo de que no se cumpla alguna de las restricciones del problema es menor o igual a la sumatoria del nivel de riesgo de cada restricción por separado:

$$\sum_{i=1}^{n} (1 - \alpha_i) \qquad (12)$$

3. APLICACIÓN Y DESARROLLO

Para esquematizar el desarrollo y aplicación de las distintas herramientas utilizadas en el presente estudio, recurrimos a un ejemplo sencillo en los siguientes términos:

Una fábrica obtiene tres tipos de producto, a partir de 6 tipos de materias primas distintos; se conocen las proporciones de esta ultimas, que se consumen en la fabricación de cada tipo de producto; por otro lado, la disponibilidad de cada materia prima es incierta; y sólo se cuenta con una pequeña muestra a modo de registro, de las cantidades que ha sido posible obtener en momentos puntuales en el pasado. Los precios de venta de los productos se conocen y se mantiene constantes durante el periodo de estudio.

Se busca encontrar las cantidades que se pueden fabricar de cada producto, con un 95% de probabilidad; teniendo en cuenta las variaciones en la cantidad disponible de cada materia prima, tal que se maximicen los ingresos de la empresa. Esto resulta de utilidad por ejemplo, en la planificación de la contratación de mano de obra, capacidad de producción y demás costes fijos; que requieren conocer el nivel mínimo de producción de antemano.

Tenemos por tanto un problema de programación lineal estocástica, donde todos los parámetros son conocidos, a excepción de los términos del lado derecho; que representan en este caso la disponibilidad futura de cada tipo de materia prima, de la que solo se puede estimar su distribución de probabilidad; basándose en un pequeña muestra, de las cantidades que ha sido posible obtener en momentos puntuales en el pasado. Suponemos que la distribución de probabilidad de la disponibilidad de cada materia prima es independiente de las demás.

Definimos:
x1: Cantidad a fabricar del producto tipo A
x2: Cantidad a fabricar del producto tipo B
x3: Cantidad a fabricar del producto tipo C

Los precios de mercado para cada tipo de producto son respectivamente de 6.1, 5.4 y 4.2 unidades monetarias.

Las proporciones de cada materia prima que se consumen en la fabricación de cada tipo de producto, están definidas por la matriz:

	Prod Tipo A	Prod Tipo B	Prod Tipo C
M.P. Tipo 1	0.09	0.27	0.22
M.P. Tipo 2	0.10	0.12	0.13
M.P. Tipo 3	0.17	0.27	0.04
M.P. Tipo 4	0.19	0.14	0.37
M.P. Tipo 5	0.24	0.08	0.14
M.P. Tipo 6	0.21	0.12	0.10

Tabla Nº 1 Matriz tecnológica

Para obtener el máximo beneficio, según la disponibilidad (estocástica) de cada materia prima; Aplicaremos el método de restricciones probabilísticas, con distribuciones de probabilidad independientes:

$$\begin{aligned}
&\textbf{Max } 6.1x1+5.4x2+4.2x3 \\
&\text{s.a} \\
&0.09\ x1 + 0.27\ x2 + 0.22\ x3 <= dMP1 \\
&0.10\ x1 + 0.12\ x2 + 0.13\ x3 <= dMP2 \\
&0.17\ x1 + 0.27\ x2 + 0.04\ x3 <= dMP3 \\
&0.19\ x1 + 0.14\ x2 + 0.37\ x3 <= dMP4 \\
&0.24\ x1 + 0.08\ x2 + 0.14\ x3 <= dMP5 \qquad (13) \\
&0.21\ x1 + 0.12\ x2 + 0.10\ x3 <= dMP6 \\
&x1,\ x2,\ x3 \geq 0
\end{aligned}$$

Donde los parámetros estocásticos dMP1, ..., dMP6 representan la disponibilidad de cada materia prima, a que la empresa puede acceder en un momento dado.

En este punto introducimos el paquete estadístico R, que servirá como herramienta de trabajo principal en nuestro análisis; se ha elegido principalmente por su versatilidad, por que cuenta con librerías especificas, que permiten abordar problemas de inferencia bayesiana y programación lineal; y por que hace posible definir funciones que incorporen en una sola herramienta, los hallazgos del presente trabajo; de tal forma que un usuario sin conocimientos profundos sobre el tema, pueda utilizar con facilidad la herramienta así formulada; en la solución de problemas similares de programación lineal estocástica. Este entorno estadístico de libre circulación, promete convertirse en el software estadístico de referencia por su amplia difusión y uso entre la comunidad académica; por que brinda una gran versatilidad al poder programar funciones específicas, por la muy amplia cantidad de herramientas con la que cuenta; la cantidad y calidad de código y funciones compartidos en la red, por un entusiasta y creciente número de desarrolladores e investigadores del campo de la estadística, la matemáticas y le ingeniería; agrupados entorno al portal Cran-r (http://cran.r-project.org/), y por su creciente compatibilidad con los principales entornos comerciales de análisis estadístico.

Como ya se había comentado, se cuenta con una pequeña muestra de las cantidades de cada materia prima; que ha sido posible obtener en momentos puntuales, los valores correspondientes se almacenan en el “fichero **reg**”**,** para ello se selecciona y copia las celdas de la hoja de cálculo que contiene los datos; junto con los nombres de las columnas, y a continuación se ejecuta la siguiente instrucción en R:

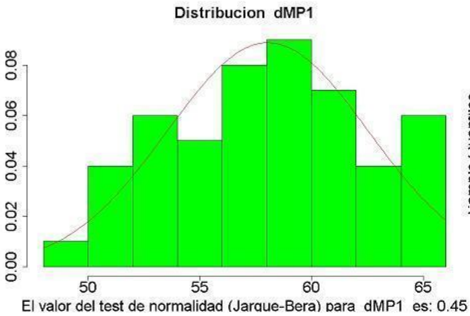
```
> reg<-read.table("clipboard", header=T)
```

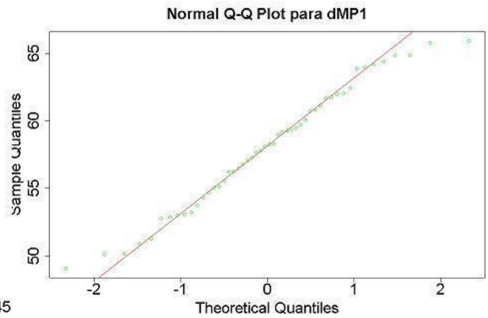
Para inferir la distribución de probabilidad de la disponibilidad de cada materia prima, debemos analizar primero la normalidad de los datos; para ello generamos un histograma, un gráfico del ajuste de los datos de la muestra a la recta normal; y el test de normalidad de Jarque-Bera para cada variable i. Este último requiere instalar y cargar con anterioridad la librería **tseries** [17] en R. (Mediante dicho test se acepta la hipótesis de normalidad de la muestra para p-value no inferiores a 0,05):

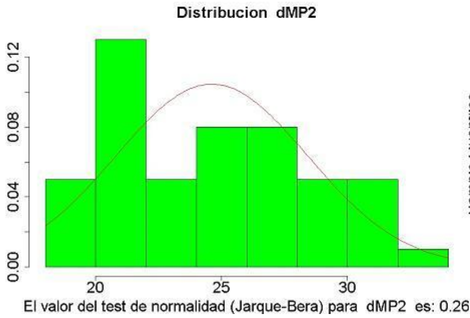
```
> k<-jarque.bera.test(reg[,i])
> k$data.name<-colnames(reg)[i]
> print(k)

> hist(reg[,i],cex.lab=2,ylab="",cex.lab=2,cex.axis=2,
xlab=paste('El valor del test de normalidad (Jarque-Bera)
para ',colnames(reg)[i],' es:',round(k$p.value,2)),main=
paste("Distribucion ",colnames(reg)[i]),cex.main=2,
col="green",freq=FALSE)ç

>curve(dnorm(x,mean=mean(reg[,i]),sd=sd(reg[,i])),col="red"
,xlab="",ylab="",add=T)

> qqnorm(reg[,i],cex.lab=2,cex.axis=2,main=paste("Normal Q-
Q Plot para",colnames(reg)[i]),cex.main=2,col="green")

> qqline(reg[,i],col="red")
```

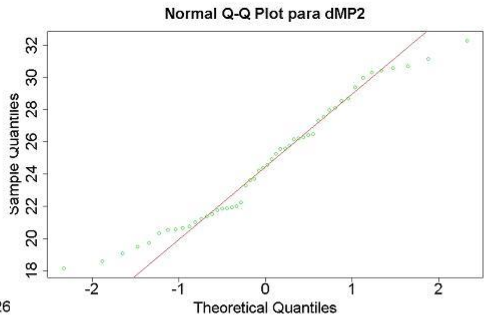
Obtenemos para cada variable los valores del test de Jarque-Bera y las gráficas correspondientes:

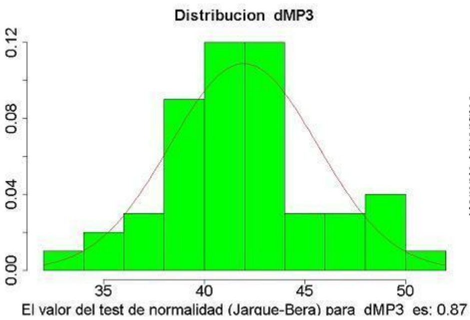
```
Jarque Bera Test
data:  dMP1 X-squared = 1.5955, df = 2, p-value = 0.4503
data:  dMP2 X-squared = 2.7152, df = 2, p-value = 0.2573
data:  dMP3 X-squared = 0.2895, df = 2, p-value = 0.8652
data:  dMP4 X-squared = 5.1007, df = 2, p-value = 0.0780
data:  dMP5 X-squared = 1.1932, df = 2, p-value = 0.5507
data:  dMP6 X-squared = 0.0111, df = 2, p-value = 0.9945
```

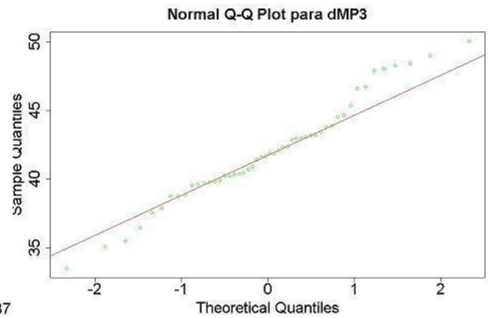

Distribucion dMP1
El valor del test de normalidad (Jarque-Bera) para dMP1 es: 0.45

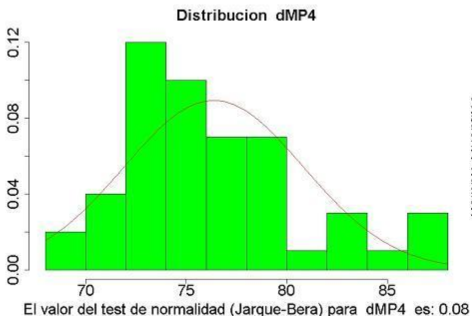

Normal Q-Q Plot para dMP1
Sample Quantiles
Theoretical Quantiles

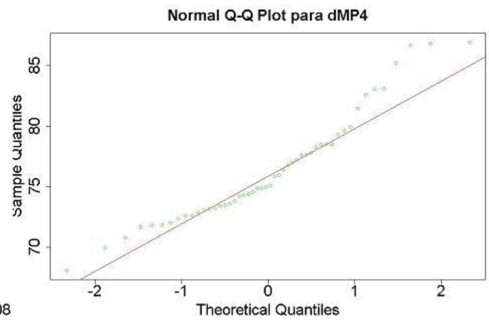

Distribucion dMP2
El valor del test de normalidad (Jarque-Bera) para dMP2 es: 0.26

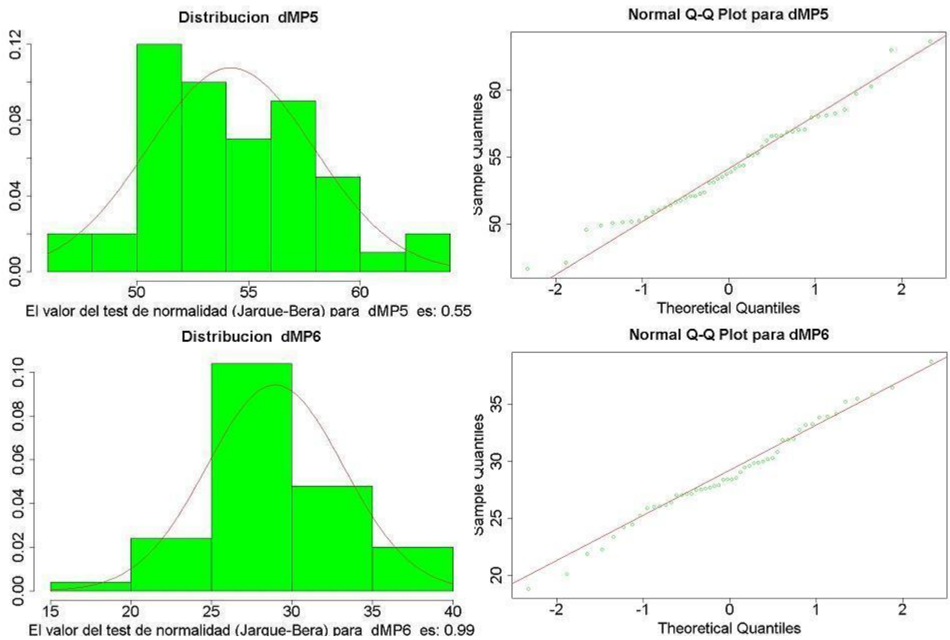

Normal Q-Q Plot para dMP2
Sample Quantiles
Theoretical Quantiles


Distribucion dMP3
El valor del test de normalidad (Jarque-Bera) para dMP3 es: 0.87


Normal Q-Q Plot para dMP3
Sample Quantiles
Theoretical Quantiles


Distribucion dMP4
El valor del test de normalidad (Jarque-Bera) para dMP4 es: 0.08


Normal Q-Q Plot para dMP4
Sample Quantiles
Theoretical Quantiles



Gráfica Nº1 Histograma y ajuste a la recta normal para cada variable

Mediante la inspección visual y la ejecución del test de Normalidad, podemos pensar que todas la series se ajustan bien a la distribución gaussiana, desconociendo la varianza poblacional, pero con un tamaño muestral relativamente grande (n>30) podemos aproximar esta por la varianza muestral $\left(\alpha^2 \approx s^2\right)$.

Podemos ahora fijar, la expresión para la distribución de cada variable estocástica de la forma:

$$N(a = \mu, b^2 = s^2) \tag{14}$$

Para ello creamos una función específica en R:

```
dp<-function(y)
{
m<-NULL;v<-NULL;n<-NULL;
for(i in(1:ncol(y)))
{
       m<-mean(y[,i]);
       v<-var(y[,i]);
       n<-colnames(y)[i];
```

```
    print(paste('La distribución de',n,'es: N( ',toString
    (round(m,2)),'  ,  ',toString(round(v,2)),'  )',sep='
    '));
}
}
```

Una vez ingresamos los datos obtenemos:

```
> dp(reg)
[1] "La distribución de dMP1 es: N(  58.04  ,  20.15  )"
[1] "La distribución de dMP2 es: N(  24.61  , 14.57  )"
[1] "La distribución de dMP3 es: N(  41.96  ,  13.41  )"
[1] "La distribución de dMP4 es: N(  76.37  ,  19.93  )"
[1] "La distribución de dMP5 es: N(  54.2  ,  13.74  )"
[1] "La distribución de dMP6 es: N(  28.96  ,  17.9  )"
```

Siguiendo con nuestro análisis, una vez contamos con la distribución de la disponibilidad de cada materia prima; recurrimos al método de restricciones probabilísticas, para obtener una solución al modelo de programación lineal estocástica planteado. Ante la imposibilidad de cumplir todas las restricciones en el 100% de los casos, se plantea hacerlo con una probabilidad determinada α; habitualmente y en nuestro caso del 95%:

Para cada restricción del tipo $a_i x \le b_i$, con F_b siendo la función de distribución de la variable estocástica b_i tenemos:

$$P(a_i x \le b_i) \ge \alpha \tag{15}$$

Siendo que:

$$P\left(a_i x \le b_i\right) = 1 - P\left(b_i \le a_i x\right) \tag{16}$$

Entonces:

$$P\left(b_i \le a_i x\right) = F_{b_i}(a_i x) \le 1 - \alpha \tag{17}$$

Donde (1-α) es el nivel de riesgo asumible de que se incumpla la restricción, con lo que tenemos que encontrar un η tal que $F_{b_i}(\eta) = 1 - \alpha$ o lo que es lo mismo $F_{b_i}^{-1}(1-\alpha) = \eta$, y así obtener restricciones lineales del tipo $a_i x \le \eta$.[9]

En nuestro caso, al existir 6 restricciones con probabilidad independiente; como hemos visto con anterioridad, cuando se busca que cada restricción de verifique con una probabilidad distinta α_i, el riesgo de que no se cumpla alguna de las restricciones del problema es menor o igual a:

$$\sum_{i=1}^{n}(1-\alpha_i) \qquad (18)$$

En nuestro caso, buscando el limite a la izquierda del 5% (que es el nivel de riesgo que habitualmente se usa en estos casos):[9]

$$\sum_{i=1}^{n}(1-\alpha_i) \leq 0.10 \qquad (19)$$

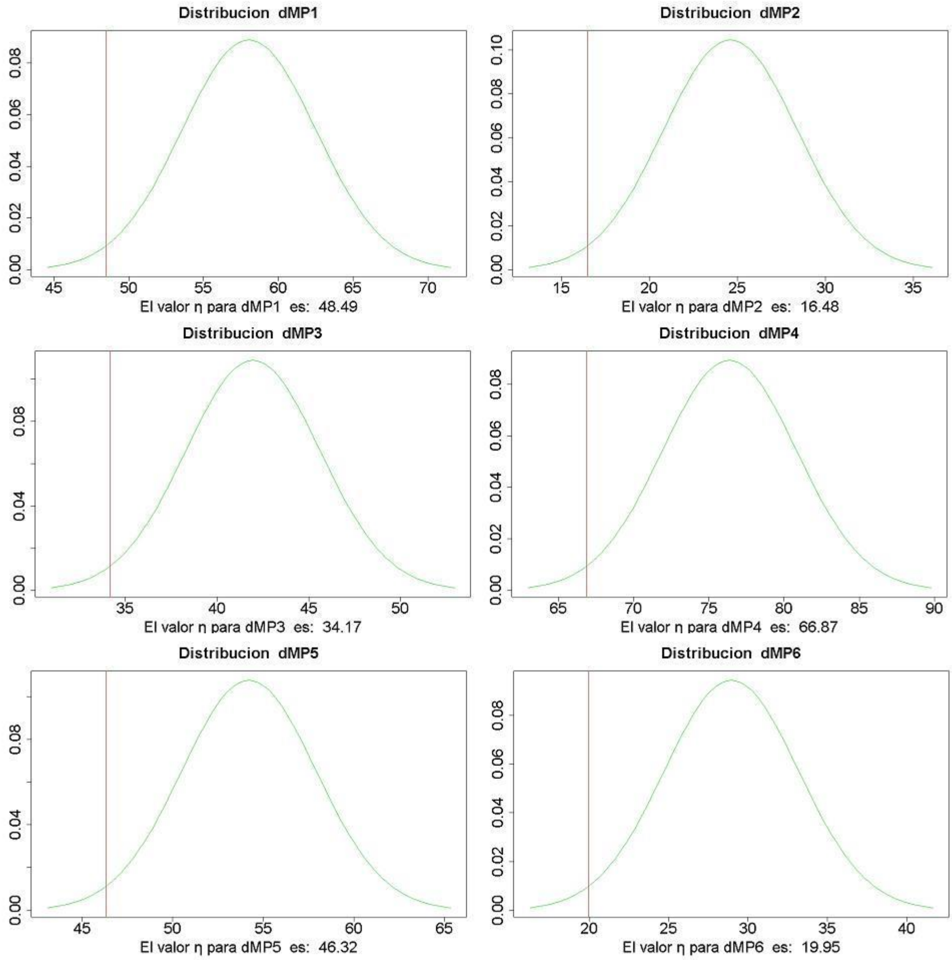
Asumiendo el mismo nivel de riesgo asumible para cada variable, esto es que todos los α_i sean iguales:

$$6*(1-\alpha_i) \leq 0.10 \qquad (20)$$

Así, fijando un nivel de riesgo asumible del 1,7% para cada variable; conseguimos que el conjunto del modelo, mantenga un nivel de riego por debajo del 5%.

En R usamos la función qnorm, para calcular el valor η dado un nivel de probabilidad (1-α)/2; en nuestro caso (1-0.9667)/2= 0.017, lo calculamos para cada variable y lo ubicamos gráficamente en la distribución de la misma:

```
> a<-mean(reg[,i])
> b<-sd(reg[,i])
> c<-a-(3*b)
> d<-a+(3*b)
> n<-qnorm(0.1/6,a,b)
> curve(dnorm(x,a,b),c,d,cex.lab=2,cex.axis=2,ylab="",
xlab=paste('El valor η para',colnames(reg)[i],' es: ',
round(n,2)),main=paste("Distribucion ",colnames(reg)[i]),
cex.main=2,col="green")
> abline(v=n,col="red")
```



Gráfica Nº 2 Percentil correspondiente al valor n para cada variable

Con estos valores volvemos al modelo general con restricciones probabilísticas, que se cumplen con un nivel de probabilidad α, donde para convertir estas restricciones probabilísticas en lineales; recurrimos a la función cuartíl de cada variable estocástica.

Como:

$$P_{\xi}(x \mid Ax \leq \xi) \geq \alpha \Leftrightarrow F_{\xi}(Ax) \leq (1-\alpha) \Leftrightarrow Ax \leq F_{\xi}^{-}(1-\alpha) \qquad (21)$$

Entonces:

Max 6.1 x1 + 5.4 x2 + 4.2x3

s.a

0.09x1 + 0.27x2 + 0.22x3 <= $F^{-1}_{dMP1}(1-\alpha)$

0.10x1 + 0.12x2 + 0.13x3 <= $F^{-1}_{dMP2}(1-\alpha)$

0.17x1 + 0.27x2 + 0.04x3 <= $F^{-1}_{dMP3}(1-\alpha)$ (22)

0.19x1 + 0.14x2 + 0.37x3 <= $F^{-1}_{dMP4}(1-\alpha)$

0.24x1 + 0.08x2 + 0.14x3 <= $F^{-1}_{dMP5}(1-\alpha)$

0.21x1 + 0.12x2 + 0.10x3 <= $F^{-1}_{dMP6}(1-\alpha)$

x1, x2, x3 ≥ 0

Los valores $\{F^{-1}_{dMP1}(1-\alpha)\ \ .\ .\ .\ \ F^{-1}_{dMP6}(1-\alpha)\}$ son los η que hemos calculado anteriormente, con lo que el modelo determinista equivalente; mediante el método de de restricciones estocásticas, que se cumplen con una probabilidad α sería:

Max 6.1 x1 + 5.4 x2 + 4.2 x3

s.a

0.09x1 + 0.27x2 + 0.22x3 <= 48.49

0.10x1 + 0.12x2 + 0.13x3 <= 16.48

0.17x1 + 0.27x2 + 0.04x3 <= 34.17 (23)

0.19x1 + 0.14x2 + 0.37x3 <= 66.87

0.24x1 + 0.08x2 + 0.14x3 <= 46.32

0.21x1 + 0.12x2 + 0.10x3 <= 19.95

x1, x2, x3 ≥ 0

Teniendo ya un modelo determinista equivalente con restricciones lineales, usamos el software de programación lineal y entera **lpSolver** [4], que cuenta con una librería especifica para su Implementación; dentro del entorno R.

Luego de instalar **lpSolver**, se hace necesario definir una serie de parámetros; la función general tiene la forma [7]:

```
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)
```

Hay una serie de valores por defecto que en nuestro caso no es necesario modificar, nos importan solo los 5 primeros:

direction = "min" o “max”, según se busque minimizar/maximizar
objective.in, vector de coeficientes de la función objetivo
const.mat, matriz de restricciones
const.dir, sentido de cada restricción (">=","<=","=",">" o "<")
const.rhs, vector del lado derecho

Incorporamos los datos del modelo determinista equivalente, con anterioridad la librería **lpSolver** debe estar instalada y cargada:

```
> f.obj<-c(6.1,5.4,4.2)
>f.con<-
matrix(c(0.09,0.27,0.22,0.1,0.12,0.13,0.17,0.27,0.04,0.19,0
.14,0.37,0.24,0.08,0.14,0.21,0.12,0.10),nrow=6,byrow=T)
> f.dir<-c("<=","<=","<=","<=","<=","<=")
> for (i in (1:ncol(reg)))
+ {
+ a<-mean(reg[,i]);
+ b<-sd(reg[,i]);
+ h[i]<-qnorm(0.1/6,a,b);
+ }
> h
[1] 48.48512 16.48423 34.16980 66.86944 46.31502 19.95118
> f.rhs<-h
> lp("max",f.obj,f.con,f.dir,f.rhs)
Success: the objective function is 786.8368
> lp("max",f.obj,f.con,f.dir,f.rhs)$solution
[1]  32.727173 105.291830   4.434557
```

Con lo que la solución para el modelo determinista equivalente, con un riesgo asumible del 5%; es producir 32.727 unidades del producto tipo A, 105.292 unidades del producto tipo B; y 4.435 unidades del producto tipo C. Se obtiene de esta forma un beneficio de 786.837 unidades monetarias, como se ha comentado anteriormente; esta solución también lo es para el modelo estocástico inicial.

4. RESULTADOS

Vamos ahora a comprobar que la solución encontrada, garantiza mantener el nivel de riesgo asumible por debajo del 5% de probabilidad; para corroborarlo modificamos la matriz de restricciones, el vector de la dirección de las restricciones y el del lado derecho; para incluir nuevas restricciones que garanticen, que se produce al menos la cantidad de cada tipo de producto;

según la solución optima obtenida (todos los valores a la derecha del valor del percentíl de la distribución):

```
>f.con2<-
matrix(c(0.09,0.27,0.22,0.1,0.12,0.13,0.17,0.27,0.04,0.19,0
.14,0.37,0.24,0.08,0.14,0.21,0.12,0.10,1,0,0,0,1,0,0,0,1),n
row=9,byrow=T)
> f.dir2<-c("<=","<=","<=","<=","<=","<=",">=",">=",">=")
> m<-lp("max",f.obj,f.con,f.dir,f.rhs)$solution
> f.rhs2<-union(f.rhs,m)
> f.rhs2
[1]  48.485123  16.484229  34.169796  66.869436  46.315018
19.951182  32.727173 105.291830   4.434557
```

Siguiendo con el proceso de comprobación, del resultado con un margen de error del 5%; simulamos diez mil vectores aleatorios del lado derecho, con igual media y varianza que las distribuciones de la muestra de cada variable.

```
> for(i in(1:ncol(reg)))
+ {
+ set.seed(round(rnorm(1)*100,0));
+ z<-rnorm(10000,mean=mean(reg[,i]),sd=sd(reg[,i]));
+ x<-union(x,z);
+ }
> u<-matrix(x,ncol=ncol(reg))
> dim(u)
[1] 10000     6
```

Tenemos ahora una matriz **u** con 10.000 simulaciones del vector del lado derecho, que siguen la distribución probabilística de cada variable; para comprobar que el nivel de probabilidad con que se incumple alguna restricción, no alcanza el 5%; creamos una rutina que evalúe uno a uno los 10 mil registros, y encuentre la proporción en que se viola alguna restriccion; en **lpSolver** el valor de retorno con extensión $Status, toma valor cero en caso de que exista solución viable para el modelo y 2 en caso contrario; aprovechamos esto para construir nuestra estadística de fallos:

```
> for (j in(1:10000))
+ {
+ for (i in(1:ncol(reg))) f.rhs2[i]<-u[j,i];
+ f[j]<-(lp("max",f.obj,f.con2,f.dir2,f.rhs2)$status);
+ }
> length(f[f==2])
[1] 479
> t<-sum(f[f==2])/200;
```

```
>   print(paste('La   probabilidad   de   error   es   del:
',round(t,2),' %'));
"La probabilidad de error es del:  4.79  %"
```

Con lo cual queda demostrado, que la solución hallada para el sistema determinista equivalente; respeta los niveles de riesgo asumible establecidos por parte del decisor en el caso estocástico, al obtener 479 fallos en diez mil simulaciones.

Suponemos ahora que por motivos técnicos y/o económicos, el decisor esta interesado en conocer; las variaciones en el beneficio para distintos niveles de riesgo asumible, si tomamos un rango de probabilidad de éxito; que va del 70% al 99% con incrementos del 1%, en nuestro ejemplo sería un riesgo asumible que decrece del 30% al 1%.

```
> q<-seq(0.7,0.99,by=.01)
> q<-(1-q)
 q
 0.30  0.29  0.28  0.27  0.26  0.25  0.24  0.23  0.22  0.21  0.20
0.19 0.18 0.17 0.16 0.15 0.14 0.13 0.12 0.11 0.10 0.09 0.08
0.07 0.06 0.05 0.04 0.03 0.02 0.01
```

Con lo cual hemos creado un vector con las probabilidades a evaluar. A continuación generamos una matriz, donde cada fila contiene los valores n para cada variable; según cada nivel de probabilidad:

```
>h<-matrix(ncol=ncol(reg),nrow=length(q))
> for (j in (1:length(q)))
+ {
+ for (i in (1:ncol(reg)))
+ {
+ a<-mean(reg[,i]);
+ b<-sd(reg[,i]);
+ h[j,i]<-qnorm(q[j]/3,a,b);
+ }
+ }
```

Con esta información podemos calcular los valores máximos de beneficio, y nivel de producción de cada producto; para cada nivel de riesgo asumible, según el modelo:

```
>r<-matrix(nrow=nrow(h),ncol=4)
> for (i in (1:nrow(r)))
+ {
+ f.rhs<-h[i,];
+ r[i,1]<-lp("max",f.obj,f.con,f.dir,f.rhs)$objval;
+ }
 for (i in (1:nrow(r)))
```

```
+ {
+ f.rhs<-h[i,];
+ for(j in (1:3))
+ {
+ r[i,j+1]<-lp("max",f.obj,f.con,f.dir,f.rhs)$solution[j];
+ }
+ }
> rownames(r)<-seq(30,1,by=-1)
> colnames(r)<-c("Ingresos","Uds P1","Uds P2","Uds P3")
```

La tabla así obtenida sirve como soporte para la toma de decisiones, según el nivel de riesgo asumible; teniendo en cuenta las consideraciones oportunas del intercambio coste/beneficio:

	BENEFICIO	P1	P2	P3
30	919.2150	39.98459	109.99030	19.3717449
29	916.2078	39.81973	109.88357	19.0324184
28	913.1234	39.65063	109.77410	18.6843849
27	909.9565	39.47701	109.66169	18.3270433
26	906.7012	39.29855	109.54616	17.9597269
25	903.3510	39.11487	109.42725	17.5816942
24	899.8984	38.92559	109.30470	17.1921169
23	896.3353	38.73025	109.17824	16.7900651
22	892.6523	38.52834	109.04752	16.3744905
21	888.8390	38.31928	108.91218	15.9442044
20	884.8833	38.10241	108.77178	15.4978515
19	880.7714	37.87699	108.62583	15.0338764
18	876.4874	37.64212	108.47378	14.5504818
17	872.0127	37.39681	108.31496	14.0455747
16	867.3257	37.13984	108.14861	13.5166971
15	862.4003	36.86982	107.97379	12.9609337
14	857.2057	36.58503	107.78942	12.3747900
13	851.7043	36.28343	107.59416	11.7540251
12	845.8498	35.96246	107.38637	11.0934193
11	839.5843	35.61897	107.16399	10.3864434
10	832.8341	35.24890	106.92441	9.6247720
9	825.5030	34.84699	106.66420	8.7975478
8	817.4620	34.40615	106.37880	7.8902197
7	808.5323	33.91660	106.06186	6.8826199
6	798.4555	33.36415	105.70421	5.7455867
5	786.8368	32.72717	105.29183	4.4345572
4	773.0253	31.96998	104.80162	2.8761099

3	755.8254	31.02702	104.19115	0.9353181
2	730.5765	30.21384	101.16149	0.0000000
1	688.2046	29.31755	94.32731	0.0000000

Tabla Nº 2 Cantidad a producir según nivel de riesgo asumible

5. CONCLUSIONES

Se ha conseguido un procedimiento de solución de problemas, de programación lineal con recursos aleatorios en R; mediante un modelo determinista equivalente, basado en desigualdades probabilísticas; y el método Simplex implementado en la librería lpSolver.

Existe disponible para su descarga, una copia del código que permite a un usuario no experto; implementar fácilmente una función en dicho entorno estadístico, que automatiza todo el proceso. Toda persona interesada no tiene más que solicitar el código a los autores.

Líneas futuras de investigación, estarían dirigidas principalmente a:

- Desarrollar un software específico que aplicando el mismo algoritmo, permita sin recurrir a R; resolver este tipo de problemas.
- Implementar una solución “Cloud” o en batch, mediante la cual será posible resolver en línea este tipo de problemas; sin instalación previa, e incluso hacer uso de la herramienta desde un Smartphone.
- Seguir el mismo proceso para el desarrollo de herramientas software, que permitan resolver problemas de programación entera, programación no lineal, programación dinámica etc.; para usuarios no expertos en temas estadísticos.

6. REFERENCIAS

[1] BEALE, E.M.L. (1955): “ON MINIMIZING A CONVEX FUNCTION SUBJECT TO LINEAR INEQUALITIES”. Admiralty Research Laboratory.

[2] BIRGE, J,R, ET AL (2010): “Introduction to Stochastic Programming”, John R..Birge, Francois Louveaux editors, 2nd_Edition.

[3] COLLOMB, A. (2004): “DYNAMIC ASSET ALLOCATION BY STOCHASTIC PROGRAMMING METHODS”.

[4] ERKELAAR, M., (2011): “INTERFACE TO LP_SOLVE V. 5.5 TO SOLVE LINEAR/INTEGER PROGRAMS”.

[5] GASSMANN, H.I.; WALLACE, S.W. (1993): “SOLVING LINEAR PROGRAMS WITH MULTIPLE RIGHT-HAND SIDES. PRICING AND ORDERING SCHEMES”.

[6] KALL P. ET AL (2010): “Stochastic Linear Programming”, Peter Kall/János Mayer editors, 2nd Edition.

[7] KONIS K., (2011): “R INTERFACE FOR LP_SOLVE”, version 5.5.2.0.

[8] KORF L.A. ;WETS, R.J.B. (1996): “AN ERGODIC THEOREM FOR STOCHASTIC PROGRAMMING PROBLEMS”.

[9] MUÑOZ MARTOS, M, (1998): “PROGRAMACIÓN ESTOCÁSTICA: ALGUNAS APORTACIONES TEÓRICAS Y COMPUTACIONALES”, Universidad Complutense de Madrid.

[10] R DEVELOPMENT CORE TEAM (2011): “R: A LANGUAGE AND ENVIRONMENT FOR STATISTICAL COMPUTING”, Version 2.13.1.

[11] R DEVELOPMENT CORE TEAM (2011): “R DATA IMPORT/EXPORT”, Version 2.13.1.

[12] SEN S., HIGLE J.L. (1999): “AN INTRODUCTORY TUTORIAL ON STOCHASTIC LINEAR PROGRAMMING MODELS”. Department of Systems and Industrial Engineering The University of Arizona.

[13] VAN DER VLERK, M.H. (2004): “INTRODUCTION TO ALGORITHMS FOR RECOURSE MODELS” University of Groningen.

[14] WETS, R.J.B. (1966): “PROGRAMMING UNDER UNCERTAINTY: THE EQUIVALENT CONVEX PROGRAM”. J. Siam.

[15] WETS, R.J.B. (1974): “STOCHASTIC PROGRAMS WITH FIXED RECOURSE: THE EQUIVALENT DETERMINISTIC PROGRAM”, SIAM REVIEW Vol. 16, No. 3, July 1974.

[16] WETS, R.J.B. (1983): “SOLVING STOCHASTIC PROGRAMS WITH FIXED RECOURSE”, STOCHASTICS Vol. 1.

[17] TRAPLETTI, A. (2012): “TSERIES PACKAGE FOR TIME SERIES ANALYSIS AND COMPUTATIONAL FINANCE”. CRAN.