Investigación de Operaciones

Tema 3

Procedimientos de Solución en Programación Lineal

Profesor:
Ricardo Caballero, M.Sc.
ricardo.caballero@utp.ac.pa

Solución gráfica para problemas de Programación Lineal

- Un problema de programación lineal que involucra sólo **dos variables de decisión** puede resolverse mediante un **procedimiento de solución gráfica**.
- Cuando existen más de dos variables, no es posible marcar la solución en una gráfica bidimensional y se debe recurrir a métodos más complejos,
- Una **solución óptima** es una solución factible que proporciona el valor más favorable de la función objetivo.
- Una **solución factible** es aquella para la que todas las restricciones se satisfacen.
- La **región factible** es la reunión de todas las soluciones factibles.
- Una **solución no factible** es una solución para la que al menos una restricción se viola.
- El **valor más favorable** significa el valor más grande si la función objetivo debe maximizarse, o el valor más pequeño si la función objetivo debe minimizarse.
- La mayor parte de los problemas tendrá nada más una solución óptima.

Métodos para obtener la solución óptima

Método de isoutilidad	Punto de esquina
Se traza una serie de rectas de isoutilidad paralelas, hasta que se encuentra la de isoutilidad máxima, es decir, aquella que tiene la solución óptima	La solución óptima se encuentra en uno de los puntos de esquina de la region factible.

Resolución por el método gráfico: Método de Punto de Esquina

La forma más fácil de resolver un pequeño problema de programación lineal tal como el de FURNITURE CITY *(ver ejemplo 1 del Tema 2)* es con el método de solución gráfico.

El procedimiento gráfico es útil sólo cuando existen dos variables de decisión (tales como un número de mesas que se deben producir, x_1, y un número de sillas que se deben producir, x_2) en el problema

Función objetivo:

$$Maximizar\ Z = 70x_1 + 50x_2$$

s.a.:

$$4x_1 + 3x_2 \leq 240$$ } Restricción de carpintería

$$2x_1 + x_2 \leq 100$$ } Restricción de pintura y barnizado

$$x_i \geq 0 \quad ; i = 1,2$$ } Restricción de no negatividad

Ejemplo: Resolución por el método gráfico Método de Punto de Esquina

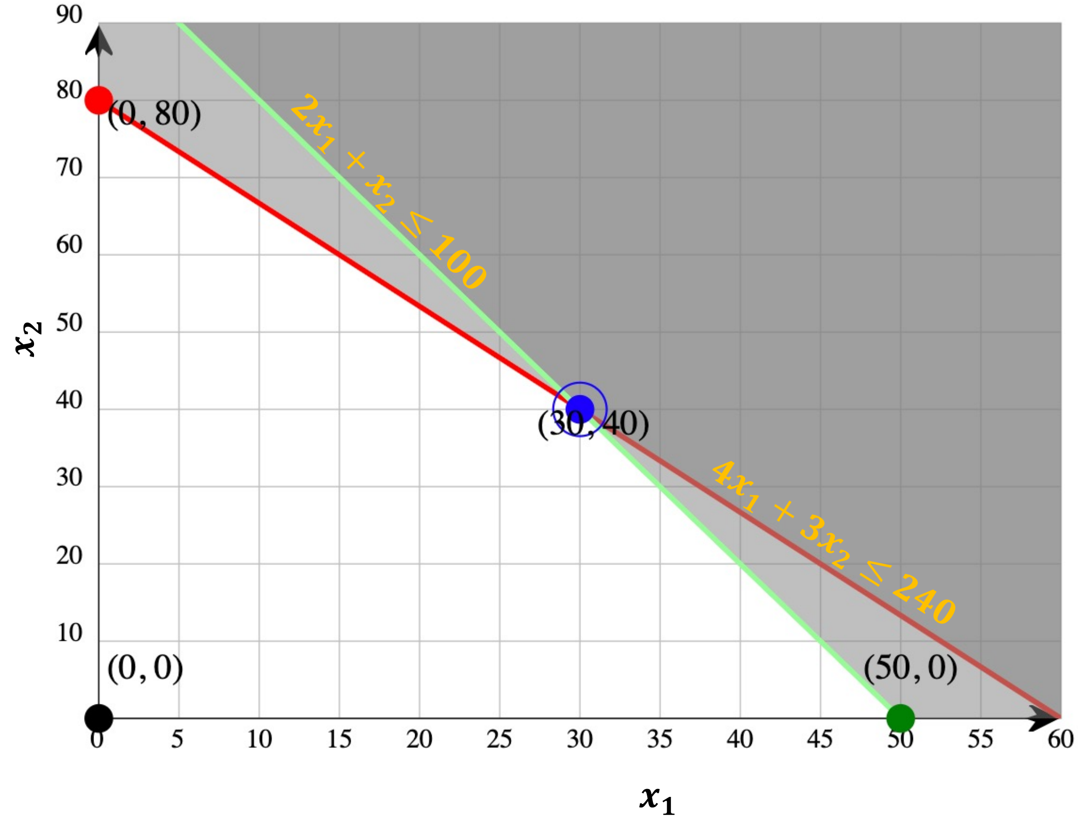
1. Graficar todas las restricciones

$$4x_1 + 3x_2 \leq 240$$

$x_1 = 0$	$x_2 = 0$
$4x_1 + 3x_2 = 240$	$4x_1 + 3x_2 = 240$
$4(0) + 3x_2 = 240$	$4x_1 + 3(0) = 240$
$3x_2 = 240$	$4x_1 = 240$
$x_2 = 80$	$x_1 = 60$

$$2x_1 + x_2 \leq 100$$

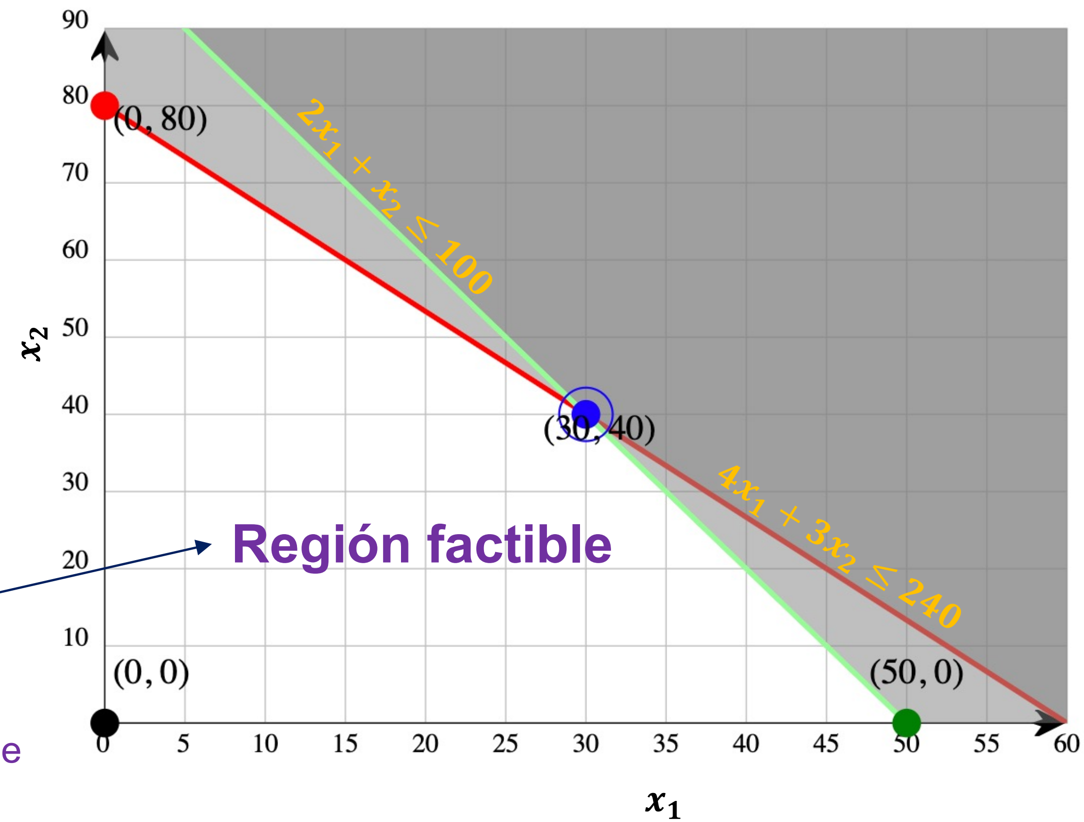
$x_1 = 0$	$x_2 = 0$
$2x_1 + x_2 = 100$	$2x_1 + x_2 = 100$
$2(0) + x_2 = 100$	$2x_1 + (0) = 100$
$x_2 = 100$	$2x_1 = 100$
	$x_1 = 50$



Ejemplo: Resolución por el método gráfico Método de Punto de Esquina

2. Definir la región factible
3. Encontrar los puntos de esquina de la región factible
4. Calcular la utilidad o costo en cada uno de los puntos de esquina factible
5. Seleccionar el punto de esquina con major valor de la función objetivo que se encontró en el paso anterior

Puntos de esquina		
x_1	x_2	Z
0	80	4000
50	0	3500
30	40	4100



Área que satisfice todas las restricciones
Cuálquier punto en esta área es una solución factible

Ejemplo: Resolución por el método gráfico Método de Punto de Esquina

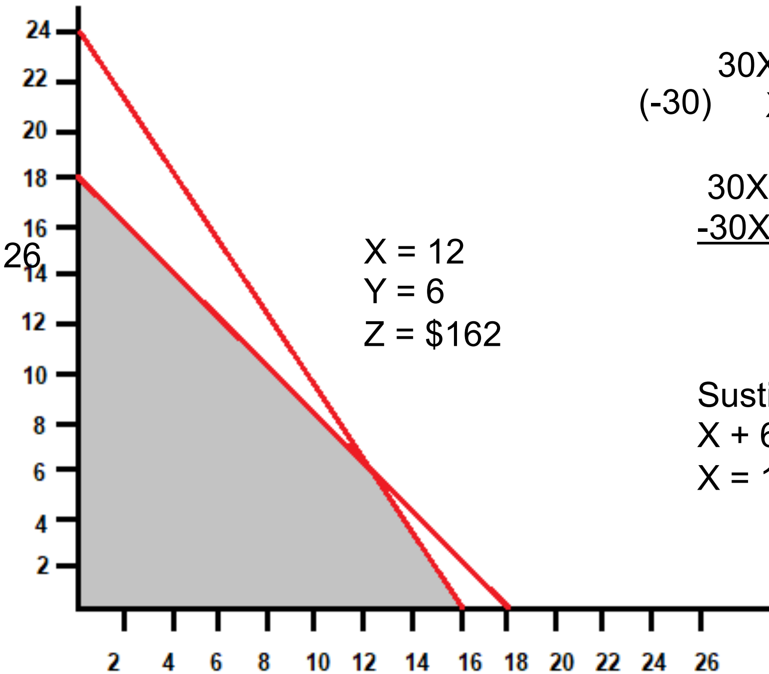
Una camioneta reparte sacos del mismo tamaño y de los tipos A y B. Los de tipo A pesan 30 kg y los B 20 kg. Por cada saco de A cobra \$10 y por cada saco de B se cobra \$7. ¿Cuántos sacos de cada clase debe transportar para maximizar ganancias si la camioneta no puede llevar más de 480 kg de estos sacos y no hay cabida para más de 18?

X – Cantidad de sacos tipo A
Y – Cantidad de sacos tipo B

$Z_{max} = 10X + 7Y$

$30X + 20Y \leq 480$
$X + Y \leq 18$

X = 0	Y = 24	X = 0	Y = 18
X = 16	Y = 0	X = 18	Y = 0



X=0
Y=18
Z = \$126

X = 12
Y = 6
Z = \$162

X=16
Z = \$160

$$30X + 20Y = 480$$
$$(-30)\quad X + Y = 18$$

$$\begin{array}{rcl} 30X + 20Y & = & 480 \\ \underline{-30X - 30Y} & = & \underline{-540} \\ -10Y & = & -60 \\ Y & = & 6 \end{array}$$

Sustituyendo:
X + 6 = 18
X = 18 – 6 = 12

Ejemplo: Resolución por el método gráfico Método de Punto de Esquina

QUIMICOS SA. es una empresa pequeña que fabrica una variedad de productos químicos. En un proceso de producción particular se utilizan tres materias primas para elaborar dos productos: un aditivo para combustible y una base para solvente. El aditivo se vende a las compañías petroleras y se utiliza en la producción de gasolina y otros combustibles. La base para solvente se vende a una variedad de compañías de productos químicos y se usa en artículos de limpieza para el hogar y la industria. Las tres materias primas se mezclan para formar el aditivo para combustible y la base para solvente, como en la tabla, en la que se muestra que una tonelada de aditivo para combustible es una mezcla de 0.4 ton de material 1 y 0.6 ton de material 3, mientras que una tonelada de base para solvente es una mezcla de 0.5 ton de material 1, 0.2 ton de material 2 y 0.3 ton de material 3.

	PRODUCTO		**Cantidad disponible para producción**
	Aditivo para combustible	Base para solvente	
Material 1	0.4	0.5	20 ton
Material 2		0.2	5 ton
Material 3	0.6	0.3	21 ton

La producción de QUIMICOS está restringida por una disponibilidad limitada de las tres materias primas.
Para el periodo de producción actual, QUIMICOS cuenta con las siguientes cantidades de cada materia prima (ver tabla)
Debido al deterioro y a la naturaleza del proceso de producción, los materiales que no se utilizan en la producción actual son inútiles y deben desecharse. El departamento de contabilidad analizó las cifras de producción, asignó todos los costos relevantes y llegó a precios para ambos productos que generarían una contribución a las utilidades de $40 por cada tonelada de aditivo para combustible producido y $30 por cada tonelada producida de base para solvente. Determine la cantidad de toneladas de aditivo para combustible y de base solvente a producir con el fin de maximizar la contribución total a las utilidades.

Ejemplo: Resolución por el método gráfico Método de Punto de Esquina

x = Número de toneladas de aditivo para combustible
y = Número de toneladas de base para solvente

F.O.: $\text{Max } Z = 40x + 30y$

Sujeto a:

$0.4x + 0.5y \leq 20$	Material 1
$0.2y \leq 5$	Material 2
$0.6x + 0.3y \leq 21$	Material 3
$x, y \geq 0$	

Ejemplo: Resolución por el método gráfico Método de Punto de Esquina

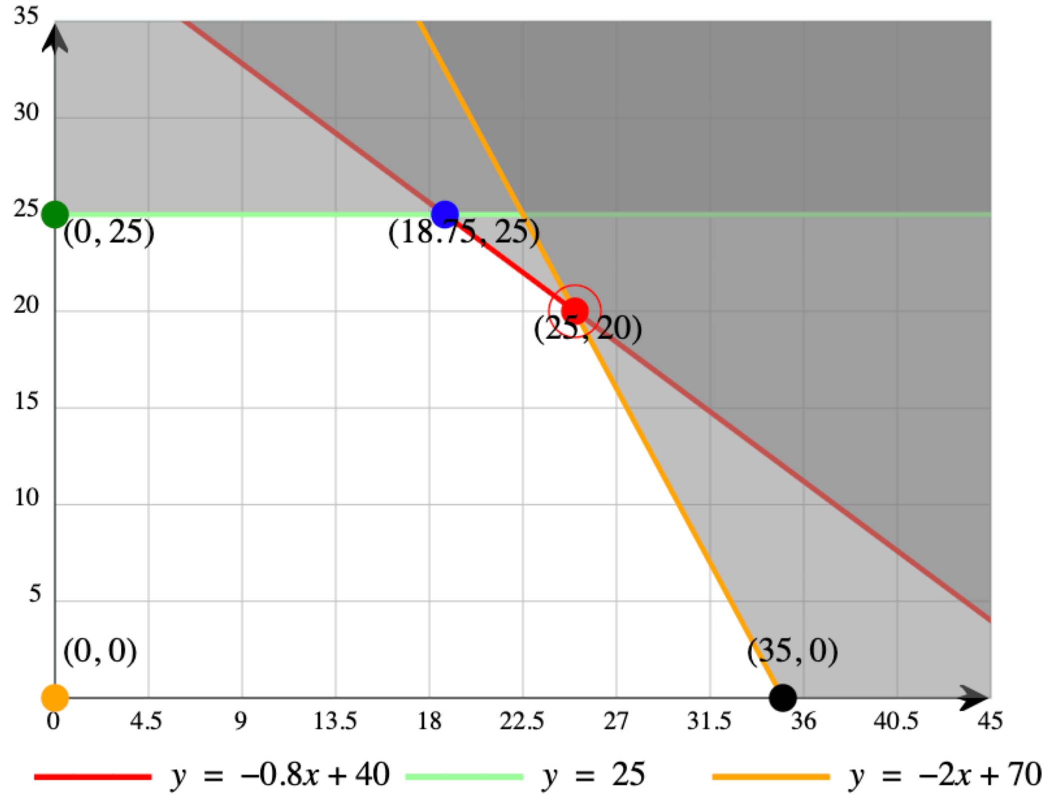
Solución

x = 25 toneladas de aditivo para combustible
y = 20 toneladas de base para solvente

Para obtener

$Z = 40\,(25) + 30(20) = 1600$ $

Vertex	Lines through vertex	Value of objective
● $(18.75, 25)$	$0.4x + 0.5y = 20$ $0.2y = 5$	1500
● $(25, 20)$	$0.4x + 0.5y = 20$ $0.6x + 0.3y = 21$	1600 Maximum
● $(0, 25)$	$0.2y = 5$ $x = 0$	750
● $(35, 0)$	$0.6x + 0.3y = 21$ $y = 0$	1400
● $(0, 0)$	$x = 0$ $y = 0$	0



Variables de Holgura (Slack)

Variable de holgura:

Variable añadida en el lado izquierdo de una restricción de menor o igual que para convertir la restricción en una igualdad. El valor de esta variable por lo general se interpreta como la **cantidad de un recurso sin utilizar.**

Solución
x = 25 toneladas de aditivo para combustible
y = 20 toneladas de base para solvente

$$\mathbf{Z = 40\,(25) + 30(20) = 1600\ \$}$$

Material 1

$0.4\,(25) + 0.5(20) \leq 20$

$\mathbf{20 \leq 20}$

Material 2

$0.2(20) \leq 5$

$\mathbf{4 \leq 5}$

Solo se utilizan 4 ton. de las 5 ton. disponibles del material 2

Material 3

$0.6\,(25) + 0.3(20) \leq 21$

$\mathbf{21 \leq 21}$

La solución óptima indica a la gerencia que la producción de 25 toneladas de aditivo para combustible y 20 toneladas de base para solvente requerirán todo el material 1 y material 3 disponibles pero sólo 4 de las 5 toneladas del material 2.

La tonelada que queda sin utilizar del material 2 se conoce como holgura.

Solución por computadora

Las soluciones de problemas lineales de más de dos variables pueden resolverse utilizando softwares o lenguajes de programación

Entre los sistemas de apoyo más comunes tenemos

POM-QM para Windows es el software de Windows más fácil de usar disponible. Este paquete de software presenta módulos separados que cubren temas tanto de la administración de operaciones como de la ciencia de la administración.

Excel QM es un complemento para Microsoft Excel que facilita el desarrollo de problemas con modelos determinísticos y probabilísticos.

Estos softwares proporciona análisis matemático para la gestión de operaciones y diversos modelos de métodos cuantitativos.

Enlace de descarga

https://wps.prenhall.com/bp_taylor_introms_11/220/56508/14466195.cw/content/

Para la solución gráfica (en el caso de MacOS)

https://www.zweigmedia.com/utilities/lpg/index.html?lang=en

Solución por computadora: Solver de Excel

La Outdoor Furniture Corporation fabrica dos productos, bancas y mesas de día de campo, que pueden ser usados en jardines de casas y parques. La firma cuenta con dos recursos principales: sus carpinteros (fuerza de mano de obra) y existencias de madera de pino para construir el mobiliario.
Durante el siguiente ciclo de producción, están disponibles 1200 horas de mano de obra según un acuerdo con el sindicato.

La firma también dispone de 3500 pies de madera de pino de buena calidad. Cada banca que Outdoor Furniture produce requiere 4 horas de mano de obra y 10 pies de madera; cada mesa de día de campo, 6 horas de mano de obra y 35 pies de madera. Las bancas terminadas redituarán una ganancia de $9 y las mesas una ganancia de $20 cada una.

¿Cuántas bancas y mesas de día de campo deberá producer Outdoor Furniture para obtener la ganancia máxima posible?

Solución por computadora: Solver de Excel

1. Escribir la función objetivo y restricciones en forma matricial (eso quiere decir cada columna representa una variable y el valor de la celda es el coeficiente
2. Escribir las funciones en el área sobrada celeste (columna D) que representarán el total de la función maximizada (celda D4) y el total de los recursos utilizados por cada restricción de mano de obra y material (celda D5 y celda D6, respectivamente). La función a utilizar es =SUMPRODUCT()

Aquí irá la solución de cantidad de bancas a producir

Aquí irá la solución de cantidad de mesas a producir

	A	B	C	D	E	F
2	Solución					
3		BANCAS	MESAS	Total		RHS
4	MAXIMIZAR	9	20	0		
5	ManoObra	4	6	0	<=	1200
6	Madera	10	35	0	<=	3500

Función objetivo

Resultado de total recursos mano de obra utilizado

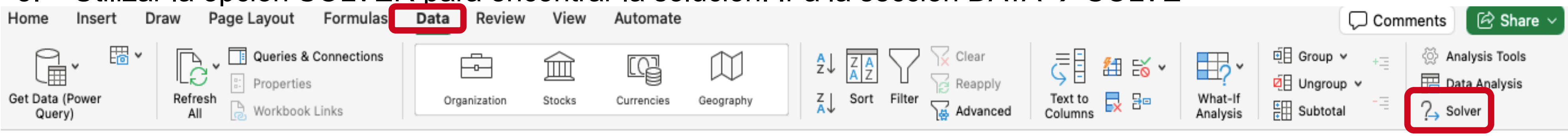
=SUMPRODUCT($B$2:$C$2,B5:C5)

Resultado de la función objetivo

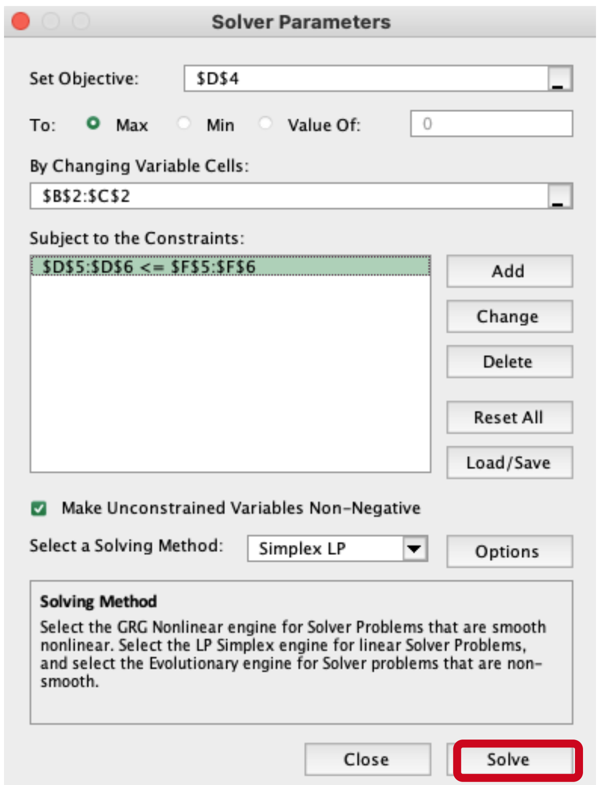
=SUMPRODUCT($B$2:$C$2,B4:C4)

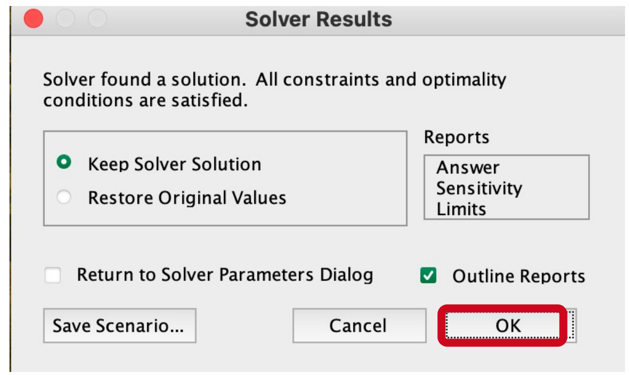
Solución por computadora: Solver de Excel

3. Utilizar la opción SOLVER para encontrar la solución. Ir a la sección DATA → SOLVE



4. Ingresar los parametros y dar click en SOLVE. Puede usar la opcion de abrir una hoja aparte con las Respuestas y una hoja aparte con el Analisis de Sensibilidad





	A	B	C	D	E	F
1						
2	Solución	262.5	25			
3		BANCAS	MESAS	Total		RHS
4	MAXIMIZAR	9	20	2862.5		
5	ManoObra	4	6	1200	<=	1200
6	Madera	10	35	3500	<=	3500

Método Simplex

- El método Simplex es un algoritmo iterativo que comienza con una solución factible inicial, avanza repetidamente hacia una solución mejor y se detiene cuando se encuentra una solución óptima y, por lo tanto, no se puede realizar ninguna mejora.
- Método de álgebra de matrices para resolver problemas de programación lineal.
- Examina sistemáticamente puntos de esquina, por medio de pasos algebraicos, hasta que se encuentra una solución óptima.

Algunos conceptos que se deben tomar en consideración:

Variable de holgura (S): Variable agregada a restricciones menores que-o-iguales-a para crear una igualdad en un método simplex. Representa una cantidad de recurso no utilizado.
Variables no básicas: Variables que no se encuentran en la mezcla de solución o base. Las variables no básicas son iguales a cero.
Fila pivote: fila correspondiente a la variable que abandonará la base para darle espacio a la variable entrante. Esta es la relación positive más pequeña que se encuentra cuando se dividen los valores de la columna de cantidad entre lso valores de la columna pivote.
Fila pivote: columna donde se encuentra el valor negativo mayor.
Número pivote: número ubicado en la intersección de la fila pivote y la columna pivote

Método Simplex: Pasos

1. **Convierta el problema de PL a la forma estándar**, lo que significa que todas las variables son no negativas y la función objetivo debe maximizarse.

Para convertir un problema de minimización en un problema de maximización, simplemente podemos multiplicar la función objetivo por −1 y luego maximizar esta función. (Recuerde que no hay restricciones de signo en el c_i). Por ejemplo, el problema de minimizar $z = 3x_1 - 5x_2$ es equivalente a maximizar $z = -3x_1 + 5x_2$. Los lados derechos negativos de las restricciones se pueden convertir en positivos multiplicando la restricción por −1 (invirtiendo el sentido de la desigualdad).

Las restricciones de igualdad no requieren modificación.

Las restricciones de desigualdad se pueden convertir en igualdades mediante la introducción de variables adicionales que componen la diferencia en los lados izquierdo y derecho de las desigualdades.

Las desigualdades menores o iguales a (≤) requieren la introducción de variables que llamaremos **variables de holgura**. Por ejemplo, una restricción como $3x_1 + 4x_2 \leq 7$ se convierte en la igualdad $3x_1 + 4x_2 + s_1 = 7$ cuando introducimos la variable de holgura s_1, donde $s_1 \geq 0$.

Las restricciones mayores o iguales a (≥) se modifican introduciendo variables excedentes . Por ejemplo, la restricción $14x_1 + 3x_2 \geq 12$ se convierte en la igualdad $14x_1 + 3x_2 - s_2 = 12$, donde s_2 es la variable excedente no negativa.

Las variables de holgura y excedente se tratarán exactamente como cualquier otra variable de decisión a lo largo del proceso de solución.

Método Simplex: Pasos

2. **Cree la solución factible inicial** estableciendo algunas de las variables en cero y encontrando los valores de las variables restantes que satisfagan las restricciones.

3. **Elija una variable no básica** (una que se establece en cero en la solución inicial) para convertirse en una variable básica (una que tiene un valor distinto de cero). Esto se hace seleccionando la variable que puede aumentar más la función objetivo, mientras sigue satisfaciendo las restricciones.

4. **Calcule la nueva solución** factible resolviendo un sistema de ecuaciones lineales que incluya la variable básica seleccionada.

5. Repita los pasos 3 y 4 hasta llegar a la solución óptima, que es el punto en el que no hay más mejoras que realizar en la función objetivo.

Ejemplo: Resolución por el método simplex

$$\begin{aligned} \text{maximize} \quad & z = 8x_1 + 5x_2 \\ \text{subject to} \quad & x_1 \le 150 \\ & x_2 \le 250 \\ & 2x_1 + x_2 \le 500 \\ & x_1,\ x_2 \ge 0 \end{aligned}$$

1. Convertir el problema en su forma estandar

$$\begin{aligned} \text{maximize} \quad & z = 8x_1 + 5x_2 + 0s_1 + 0s_2 + 0s_3 \\ \text{subject to} \quad & x_1 + s_1 = 150 \\ & x_2 + s_2 = 250 \\ & 2x_1 + x_2 + s_3 = 500 \end{aligned}$$

(Se asignan coeficientes cero a las variables de holgura en la función objetivo porque las variables de holgura no contribuyen a z).

Ejemplo: Resolución por el método simplex

2. Crear solución factible inicial
3. Elija una variable no básica

Las restricciones constituyen un sistema de m = 3 ecuaciones en n = 5 incógnitas.
Para obtener una solución factible básica inicial, necesitamos seleccionar n − m = 5 − 3 = 2 variables como variables no básicas

Podemos ver fácilmente en este caso que al elegir las dos variables x_1 y x_2 como variables no básicas y establecer sus valores en cero, no se requiere ningún cálculo significativo para resolver las tres variables básicas: $s_1 = 150$, $s_2 = 250$ y $s_3 = 500$. El valor de la función objetivo en esta solución es 0.

De hecho, una solución inicial es así de fácil de obtener cuando tenemos m variables, cada una de las cuales tiene un coeficiente de uno en una ecuación y cero coeficientes en todas las demás ecuaciones (un vector unitario de coeficientes), y cada ecuación tiene tal variable con un coeficiente de uno en ella. Por lo tanto, cada vez que se agrega una variable de holgura a cada restricción, podemos elegir todas las variables de holgura como las m variables básicas, establecer las variables restantes (n − m) en cero, y los valores iniciales de las variables básicas están simplemente dados por las constantes b en el lado derecho de las restricciones.

Basis	z	x_1	x_2	s_1	s_2	s_3	Solution
z	1	−8	−5	0	0	0	0
s_1	0	1	0	1	0	0	150
s_2	0	0	1	0	1	0	250
s_3	0	2	1	0	0	1	500

Tablau inicial

Ejemplo: Resolución por el método simplex

Tablau inicial

Basis	z	x_1	x_2	s_1	s_2	s_3	Solution
z	1	−8	−5	0	0	0	0
s_1	0	1	0	1	0	0	150
s_2	0	0	1	0	1	0	250
s_3	0	2	1	0	0	1	500

La primera columna enumera las variables básicas actuales.

La segunda columna muestra que z es (y siempre será) una variable básica; y debido a que estos elementos nunca cambiarán, realmente no es necesario mantenerlos explícitamente en el cuadro.

Las siguientes cinco columnas son los coeficientes de restricción de cada variable.

Y la última columna es el vector solución (B_j); es decir, los valores de las variables básicas.

Usando esta representación de una solución actual, ahora podemos describir el propósito y la función de cada iteración del método Simplex para un problema de maximización.

Ejemplo: Resolución por el método simplex

Tableau inicial

columna pivote

fila pivote

Elemento pivote

Basis	z	x_1	x_2	s_1	s_2	s_3	Solution
z	1	−8	−5	0	0	0	0
s_1	0	1	0	1	0	0	150
s_2	0	0	1	0	1	0	250
s_3	0	2	1	0	0	1	500

Tableau 1

Basis	z	x_1	x_2	s_1	s_2	s_3	Solution
z	1	0	−5	8	0	0	1200
x_1	0	1	0	1	0	0	150
s_2	0	0	1	0	1	0	250
s_3	0	0	$\boxed{1}$	−2	0	1	200

Se selecciona el menor elemento de la fila Zj y allí estará la columna pivote. También me indicará la variable que entra.

Se dividirá los valores Bj entre los valores de la columna pivote para cada variable y se seleccionará el menor.

Esa será la fila pivote. La cantidad donde se intersectan la fila y la columna será el elemento pivote. La fila pivote me indicará la variable que sale

Entra “X_1” y sale “S_1”

Luego el elemento pivote que en este caso es “1” hay que convertirlo en “1” dividiendo toda la fila por el número que lo convierte en “1”

Convertir todos los elementos que está arriba y debajo del “1” en “0”

Ejemplo: Resolución por el método simplex

Tableau 2

Basis	z	x_1	x_2	s_1	s_2	s_3	Solution
z	**1**	**0**	**0**	**–2**	**0**	**5**	**2200**
x_1	0	1	0	1	0	0	150
s_2	0	0	0	2	1	–1	50
x_2	0	0	1	–2	0	1	200

Tableau 3

Basis	z	x_1	x_2	s_1	s_2	s_3	Solution
z	**1**	**0**	**0**	**0**	**1**	**4**	**2250**
x_1	0	1	0	0	–1/2	1/2	125
s_1	0	0	0	1	1/2	–1/2	25
x_2	0	0	1	0	1	0	250

$x_1 = 125$

$x_2 = 250$

$$z^* = 8x_1 + 5x_2 = 8(125) + 5(250) = 2250$$

Se repite el proceso hasta que los valores de la fila Zj sean mayores o iguales que “0”, esto quiere decir, valores no negativos

Debido a que la fila de los coeficientes de la función objetivo son no negativos se ha encontrado la solución óptima con las variables de decision
$x_1 = 125$ y $x_2 = 250$, con un valor de Z max de 2250

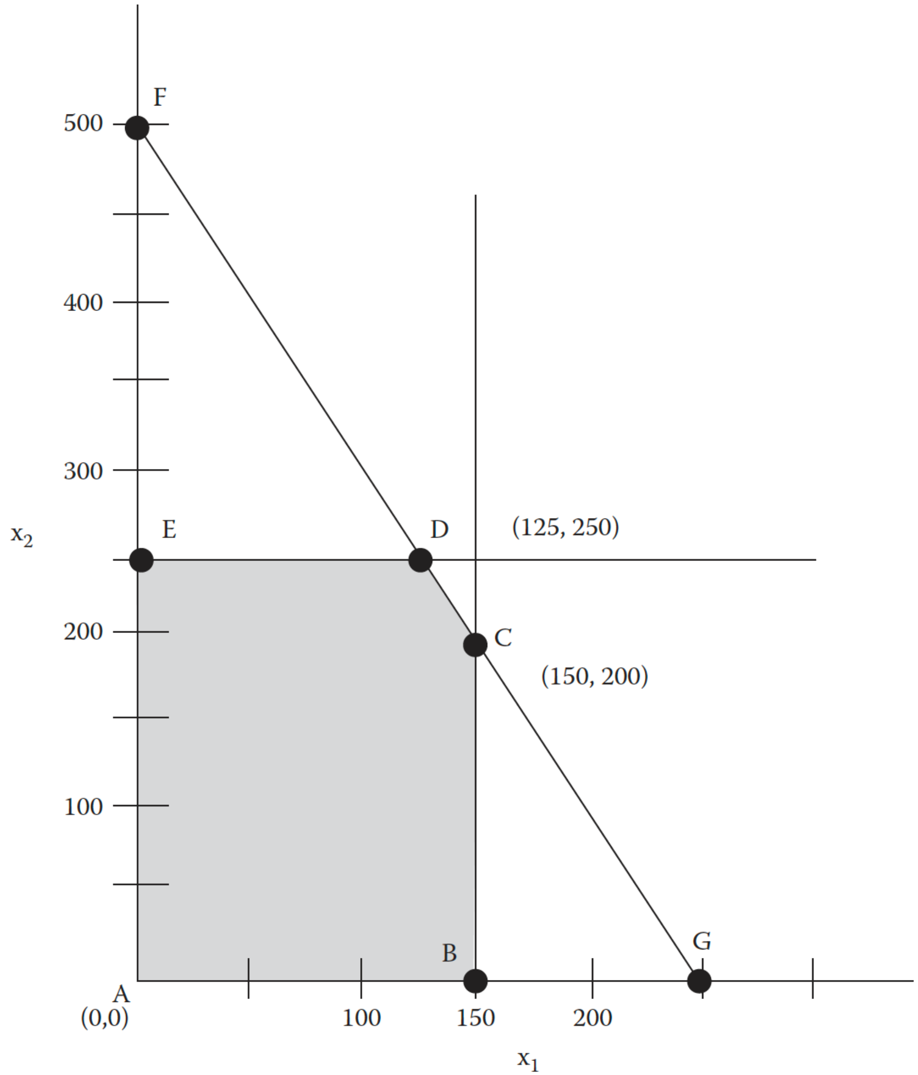
Los valores de las variables de holgura en el nivel óptimo también proporcionan información útil.

La variable de holgura s_1 para la primera restricción tiene un valor de 25, lo que indica que hay una diferencia de 25 en los lados derecho e izquierdo de la restricción; por lo tanto, $x_1 = 125$ es 25 menos que 150. (Por lo general, esto puede interpretarse como que algún recurso correspondiente a la restricción 1 no se consume por completo en el estado óptimo; tal restricción a veces se denomina restricción no vinculante).

Dado que s_2 y s_3 son no básicos y, por lo tanto, tienen un valor de cero, podemos ver que la segunda y la tercera restricción se cumplen como igualdades. (Estos recursos se utilizan al máximo de su capacidad y estas restricciones a veces se denominan restricciones vinculantes).

Ejemplo: Resolución por el método simplex

Representación gráfica



Ejemplo: Resolución por el método simplex

$$Maximizar\ Z = 70T + 50C$$

$$4T + 3C \leq 240$$
$$2T + C \leq 100$$

$$-70T - 50C + Z = 0$$

$$4T + 3C + S_1 = 240$$
$$2T + C + S_2 = 100$$

Tableau 1

Variable de decisión	T	C	S_1	S_2	B_j	
S_1	4	3	1	0	240	240/4 = 60
S_2	2	1	0	1	100	100/2 = 50
	-70	-50	0	0	0	

Fila pivote

Elemento pivote

Columna pivote

Ejemplo: Resolución por el método simplex

Tableau 2

Fila pivote → S_1

Variable de decisión	T	C	S_1	S_2	B_j	
S_1	0	1	1	-2	40	40/1 = 40
T	1	½	0	½	50	50 / ½ = 100
	0	-15	0	35	3500	

Columna pivote ↑ (C)

Tableau 3

Variable de decisión	T	C	S_1	S_2	B_j
C	0	1	1	-2	40
T	1	0	- ½	3/2	30
	0	0	15	5	4100

SOLUCIÓN ÓPTIMA

$T = 30$

$C = 40$

$MaxZ = 4100$

Ejemplo: Resolución por el método simplex

Una camioneta reparte sacos del mismo tamaño y de los tipos A y B. Los de tipo A pesan 30 kg y los B 20 kg. Por cada saco de A cobra \$10 y por cada saco de B se cobra \$7. ¿Cuántos sacos de cada clase debe transportar para maximizar ganancias si la camioneta no puede llevar más de 480 kg de estos sacos y no hay cabida para más de 18?

X – Cantidad de sacos tipo A
Y – Cantidad de sacos tipo B

$Zmax = 10X + 7Y$

$30X + 20Y \leq 480$
$X + Y \leq 18$

Se convierten las desigualdades en ecuaciones agregándoles variables de holguras y se iguala la función objetivo a cero.

$$
\begin{aligned}
30X + 20Y + S_1 \quad &= 480 \\
X + Y \quad + S_2 &= 18 \\
Z - 10X - 7Y \quad &= 0
\end{aligned}
$$

Tablón inicial del método simplex

Variable de decisión	X	Y	S_1	S_2	B_j
S_1	30	20	1	0	480
S_2	1	1	0	1	18
Z_j - C_j	-10	-7	0	0	0

El objetivo es que todos los elementos de la fila Zj-Cj sean iguales o mayores que "0"

Ejemplo: Resolución por el método gráfico

Variable de decisión	X	Y	S_1	S_2	Bj
S_1	30	20	1	0	480
S_2	1	1	0	1	18
Zj - Cj	-10	-7	0	0	0

480/30=16
18/1 = 18

2. Se dividirá los valores Bj entre los valores de la columna pivote para cada variable y se seleccionará el menor. Esa será la fila pivote. La cantidad donde se intersectan la fila y la columna será el elemento pivote. La fila pivote me indicará la variable que sale.

1. Se selecciona el menor elemento de la fila Zj-Cj y allí estará la columna pivote.
También me indicará la variable que entra.

3. Entra "X" y sale "S1" luego el elemento pivote que en este caso es "30" hay que convertirlo en "1" dividiendo toda la fila por el número que lo convierte en "1"

Variable de decisión	X	Y	S_1	S_2	Bj
X	1	2/3	1/30	0	16
S_2	0	1/3	-1/30	1	2
Zj - Cj					

4. Convertir todos los elementos que está arriba y debajo del "1" en "0"

1	1	0	1	18
-1	-2/3	-1/30	0	-16
0	1/3	-1/30	1	2

Ejemplo: Resolución por el método gráfico

Tablón 1 del Método Simplex

Variable de decisión	X	Y	S_1	S_2	Bj
X	1	2/3	1/30	0	16
S_2	0	1/3	-1/30	1	2
Zj - Cj	0	-1/3	1/3	0	160

-10	-7	0	0	0
10	20/3	1/3	0	160
0	-1/3	1/3	0	160

5. Se selecciona el menor elemento de la fila Zj – Cj para determinar la nueva columna pivote y la variable que entra

Variable de decisión	X	Y	S_1	S_2	Bj
X	1	2/3	1/30	0	16
S_2	0	1/3	-1/30	1	2
Zj - Cj	0	-1/3	1/3	0	160

16/(2/3) = 24
2/(1/3) = 6

6. Se dividen los elementos bj entre los de la columna pivote y se selecciona el menor allí estará la fila pivote y la variable que sale. El nuevo elemento pivote será 1/3. Por lo que se divide toda esa fila entre 1/3 para convertir ese elemento en 1.

Variable de decisión	X	Y	S1	S2	Bj
X					
S_2	0	1	-1/10	3	6
Zj - Cj					

Ejemplo: Resolución por el método gráfico

Variable de decisión	X	Y	S_1	S_2	Bj
X	1	2/3	1/30	0	16
S_2	0	1/3	-1/30	1	2
Zj - Cj	0	-1/3	1/3	0	160

Tablón 2 del Método Simplex

Variable de decisión	X	Y	S_1	S_2	Bj
X	1	0	1/10	-2	12
Y	0	1	-1/10	3	6
Zj - Cj	1	0	1/10	1	162

Todos los elementos de la fila Zj – Cj son iguales o mayores que cero.

7. Se convertirán en "0" todos los elementos que están por arriba y por debajo del "1".

1	2/3	1/30	0	16	para la primera fila
0	-2/3	1/15	-2	-4	
1	0	1/10	-2	12	

0	-1/3	1/3	0	160	para la primera fila
0	1/3	-1/30	1	2	
1	0	1/10	1	162	

Fundamentos básicos: R y RStudio

R es un lenguaje de programación y un entorno de software libre para la computación estadística y gráficos respaldados por la Fundación R para Computación Estadística.

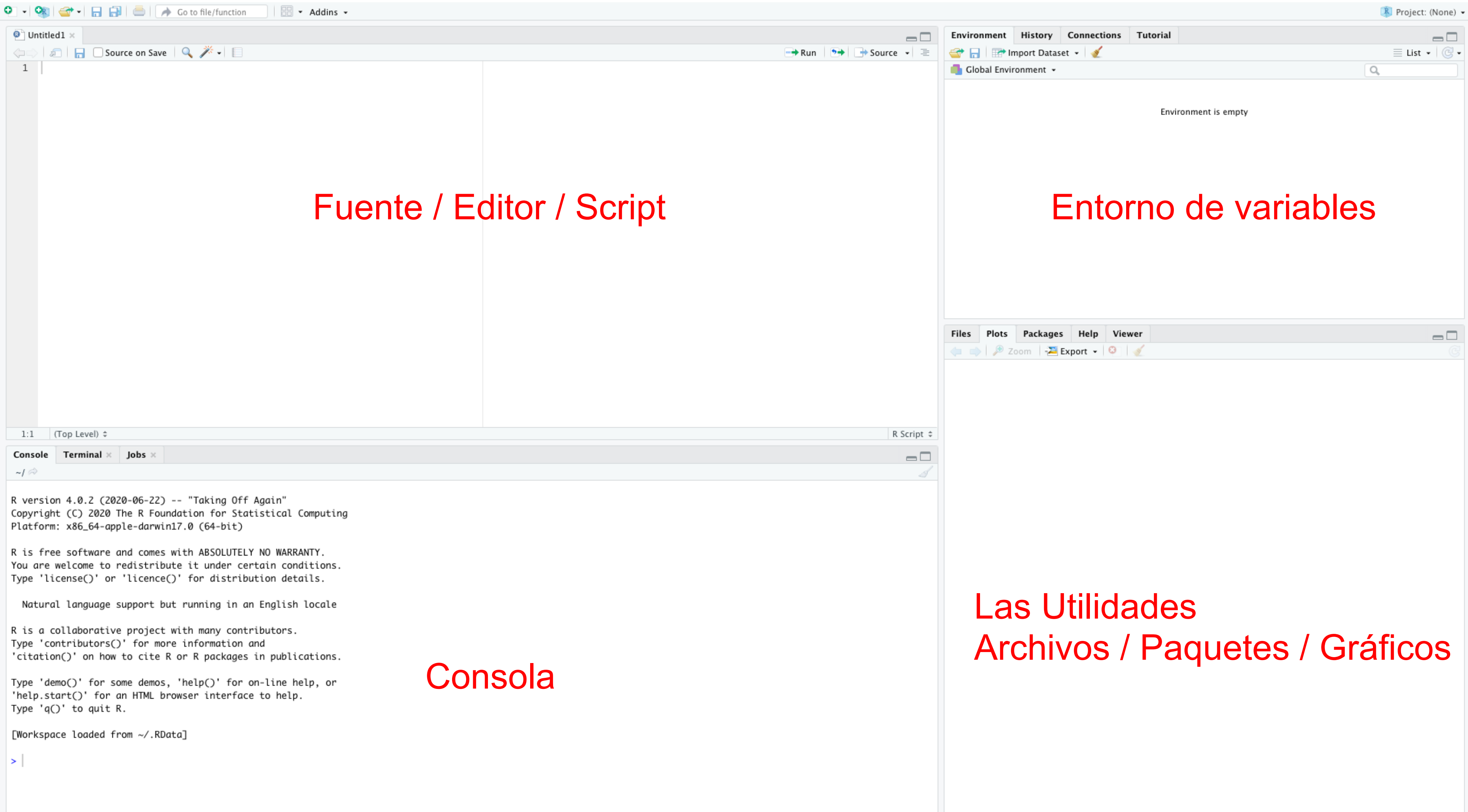
- Lenguaje de programación enfocado en el análisis gráfico y análisis estafístico

- R Studio es un interfaz gráfico para el usuario que permite
  - Escribir, editar y guardar código
  - Generar, ver gráficos
  - Administrar archivos, objetos y conjunto de datos

Para descargar R y Rstudio utilizar los siguientes links:

https://www.r-project.org/
https://rstudio.com/products/rstudio/download/

R Studio y sus partes



Programar con R: Definición de variable e impresión

<- Operador de asignación: asigna un valor a un símbolo

Print() Imprime el valor de una variable

Ejemplo:

Objeto tipo numérico considerado un vector

```
x <- 1
print(x)
[1] 1
```

O también puede escribir

```
x <- 1
x
[1] 1
```

No necesariamente se debe usar print para imprimir el valor. R guarda el valor de la variable con <- y al escribir la variable y dar click en ENTER se imprime el valor previamente guardado

Programar con R: Definición de variable e impresión

<-	Operador de asignación: asigna un valor a un símbolo
Print()	Imprime el valor de una variable

Ejemplo:

Objeto tipo caracter considerado un vector →

```
mensaje <- "hola"
mensaje
[1] "hola"
```

Se utiliza “” para escribir objeto tipo caracter o string

Programar con R: Uso de comentarios

Se utiliza para agregar comentarios

Ejemplo:

```
x <- 5        #se asigna variable
x             # se imprime automáticamente
[1] 1
```

Vector

Número

El [1] indica que x es un vector y que 5 es el primer elemento del vector

Programar con R: Uso del operador

: Se utiliza para crear una secuencia de valores x enteros

Ejemplo:

```
x <- 1:20
x
[1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

Secuencia de 1 hasta 20

Objetos y atributos

Todas las cosas que se escriban o manipulen en R se denominan **objetos**

Los objetos pueden contener cualquier tipo de data como:

- Character (string)
- Numerico (números reales)
- Entero (integer)
- Complejo
- Lógico (True/False)

El objeto más basico es un vector

→ **Un vector solamente puede contener objetos de una misma clase**

Excepción → Listas (contienen conjunto de números, caracteres, etc)

Atributos

- Nombres
- Dimensiones (matrices y arreglos)
- Clases
- Length
- Otros

Vectores y listas

C() Función utilizada para crear vectores de objetos
c – concatenar = union de objetos

Ejemplo:

```
x <- c(5,6)
x
[1] 5 6
```

← Concatenar objetos numéricos de valor 5 y 6

← Vector 5 y 6

Vectores y listas

C() Función utilizada para crear vectores de objetos
c – concatenar = union de objetos

Ejemplo:

```
x <- c(5,6)
x
[1] 5 6
```

Concatenar objetos numéricos de valor 5 y 6

Vector 5 y 6

Ejemplo:

El valor por default es 0

```
x <- vector("numeric", length = 10)
x
[1] 0 0 0 0 0 0 0 0 0 0
```

Matrices

- Utilizan un tipo especial de vector
- Son vectores con **atributos dimensionales**

Un atributo dimensional es en sí un vector integer (entero) de longitude (length) 2
Esto quiere decir (nrow, ncol)

matrix(nrow = , ncol =) Define una matriz con n cantidad de filas y columnas

Ejemplo:

```
m <- matrix(nrow = 2, ncol = 3)
m
     [,1] [,2] [,3]
[1,]   NA   NA   NA
[2,]   NA   NA   NA
```

NA significa valor no asignado (Not Assigned)

[,]

Número de fila   Número de columna

Matrices: dim() y attributes()

dim()	Establece la dimensión de un objeto
attributes()	Utilizado para obtener todos los atributos de los datos

Ejemplo:

```
attributes(m)
$dim
[1] 2 3
```

```
dim(m)
[1] 2 3
```

$ = Subconjunto de lista, binario

Matrices

matrix(nrow = , ncol =) Define una matriz con n cantidad de filas y columnas

Ejemplo:

```
m <- matrix(1:6, nrow=2, ncol=3)
m
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

También se pueden crear matrices creando atributos dimensionales

```
m <- 1:10
dim(m) <- c(2,5)
m
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
```

Matrices: cbind y rbind

cbind Es una función utilizada se usa para combinar vectores, matrices o conjunto de datos por columnas

rbind Es una función utilizada se usa para combinar vectores, matrices o conjunto de datos por filas

Ejemplo:

```
x <- 1:3
y <- 10:12

cbind(x,y)

     x  y
[1,] 1 10
[2,] 2 11
[3,] 3 12
```

```
x <- 1:3
y <- 10:12

rbind(x,y)

  [,1] [,2] [,3]
x    1    2    3
y   10   11   12
```

Matrices: byrow=TRUE

byrow=TRUE
indica que **la matriz debe ser llenada por filas**.

byrow=FALSE
indica que la matriz debe ser llenada por columnas (predeterminado).

Data.frame

La función data.frame() crea marcos de datos, colecciones de variables estrechamente acopladas que comparten muchas de las propiedades de las matrices y las listas, utilizadas como la estructura de datos fundamental por la mayoría del software de modelado de R.

```
data.frame(…, row.names = NULL, check.rows = FALSE,
           check.names = TRUE, fix.empty.names = TRUE,
           stringsAsFactors = default.stringsAsFactors())
```

Uso de data.frame

Defina los valores para las siguientes variables: nombre, fecha de nacimiento, sexo, edad. Utilice cuatro valores para cada variable y concatenelos en una tabla

```
nombre <- c("Maria","Luis","Juan","Carla")
fecha_nacimiento <- c("26/09/1990", "13/02/2000", "15/04/1993", "20/05/1995")
sexo <- c("F", "M", "M", "F")
edad <- c("31", "22", "28", "26")
estudiante <- data.frame(nombre,fecha_nacimiento, sexo, edad)

estudiante
  nombre fecha_nacimiento sexo edad
1  Maria       26/09/1990        F   31
2   Luis       13/02/2000        M   22
3   Juan       15/04/1993        M   28
4  Carla       20/05/1995        F   26
```

Como se filtran los datos de una tabla

```
 nombre fecha_nacimiento sexo edad
1  Maria     26/09/1990    F   31
2   Luis     13/02/2000    M   22
3   Juan     15/04/1993    M   28
4   Carla    20/05/1995    F   26
```

estudiante[2,3] ← Filtrar valor de la segunda fila tercera columna
[1] "M”

estudiante[,3] ← Filtrar valores de la tercera columna
[1] "F" "M" "M" "F”

estudiante$edad ← Filtrar valores del subconjunto de lista o columna edad
[1] "31" "22" "28" "26”

estudiante$edad[3] ← Filtrar tercer valor de la columna edad
[1] "28"

Limpiar la consola

Sí deseas limpiar todo el código escrito en la consola e iniciar desde cero utiliza

Ctrl + L

LpSolve

Paquete utilizado para resolver problemas de programación lineal y programación entera

Los **paquetes** también se pueden instalar desde la línea de comandos de R.
Este es un enfoque más general que funcionará en todos los entornos.

La instalación del paquete requiere un solo comando:

El paquete lpSolve R:

install.packages("lpSolve")

Instalar el paquete no es suficiente. También debe cargarse en el espacio de memoria R antes de que pueda usarse. Esto se puede hacer con el siguiente comando:

library ("lpSolve")

Para más información puede referirse al siguiente link
https://www.rdocumentation.org/

Argumentos del paquete LpSolve

- **direction**
  indica la dirección de la optimización: "mín."(por default) o "máx."

- **objective.in**
  Vector numérico de los coeficientes de la función objetivo

- **const.mat**
  Matriz de los coeficientes numéricos de las restricciones, una fila por restricción, una columna por variable (a menos que use transpose.constraints = FALSE).

- **const.dir**
  Vector de cadena de caracteres que dan la dirección de la restricción: cada valor debe ser uno de los siguientes "<," "<=," "=," "==," ">," o ">=".

- **const.rhs**
  Vector de valores numéricos para la parte derecha de las restricciones (los recursos)

Para más información puede referirse al siguiente link
https://www.rdocumentation.org/

Ejemplo 1: **Problema de maximización con R**

Dada la siguiente formulación

$$Max\ f(x,y) = 2x + 3y$$
$$x + y \leq 3$$
$$x + y \geq 0$$

Resuelve el problema de PL utilizando R

Referencia

https://www.supplychaindataanalytics.com/es/resolver-un-problema-de-programacion-lineal-simple-usando-lpsolve-en-r/

Ejemplo 1: Problema de maximización con R

```
library(lpSolve)
f.obj <- c(2,3)                                   # vector de coeficiente de la función objetivo
f.con <- matrix(c(1,1),nrow=1,byrow=TRUE)          # matriz de coeficientes para la matriz de restricciones
f.dir <- c("<=")                                   # vector de dirección de la restricción
f.rhs <- c(3)                                      # valores de la restricción
solution <- lp("max",f.obj,f.con,f.dir,f.rhs)
solution
solution$solution
```

```
Success: the objective function is 9

solution$solution
[1] 0 3
```

Referencia

https://www.supplychaindataanalytics.com/es/resolver-un-problema-de-programacion-lineal-simple-usando-lpsolve-en-r/

Ejemplo 2: Problema de maximización con R

Furniture City fabrica mesas y sillas de bajo precio. El proceso de fabricación de cada producto se parece en que ambos requieren cierto número de horas de trabajo de carpintería, así como cierto número de horas de trabajo en el departamento de pintura y barnizado. Cada mesa requiere de 4 horas de carpintería y 2 horas en el taller de pintura y barnizado. Cada silla requiere de 3 horas de carpintería, y 1 hora en la pintura y barnizado. Durante el periodo de producción actual, hay 240 horas de tiempo de carpintería disponibles, así como 100 horas de tiempo disponibles en pintura y barnizado. Cada mesa vendida genera una utilidad de $70; cada silla fabricada se vende con una utilidad de $50.

El problema de Furniture City es determinar la mejor combinación posible de mesas y sillas a fabricar, con la finalidad de alcanzar la utilidad máxima. La empresa desea que esta situación de mezcla de producción se formule como un problema de programación lineal.

Función objetivo: $$Maximizar\ Z = 70x_1 + 50x_2$$

s.a.:

$$4x_1 + 3x_2 \leq 240$$
$$2x_1 + x_2 \leq 100$$
$$x_i \geq 0 \qquad ; i = 1,2$$

Ejemplo 2: Problema de maximización con R

```
library(lpSolve)
f.obj <- c(70,50)                                    # vector de coeficiente de la función objetivo
f.con <- matrix(c(4,3,2,1),nrow=2,byrow=TRUE)        # matriz de coeficientes de restricciones
f.dir <- c("<=")                                     # vector de dirección de la restricción
f.rhs <- c(240,100)                                  # valores de la restricción
solution <- lp("max",f.obj,f.con,f.dir,f.rhs)
solution
solution$solution
```

```
Success: the objective function is 4100

solution$solution
[1] 30 40
```

Libros de referencia

- Render, B. (2016). Métodos cuantitativos para los Negocios. Editorial Pearson.
- Gohout, W. (2013). Operations Research. Oldenbourg Verlag München
- Taha, H. (2011). Investigación de Operaciones. Editorial Pearson.
- Hillier, F. & Lieberman, G. (2015). Investigación de Operaciones. McGraw-Hill
- Winston, W. (2004). Operations Research Applications and Algorithms. Thomson Brooks/Cole
- Anderson, D. & Sweeny, D. (2019). Métodos Cuantitativos para los Negocios. Cengage
- Eppen, D. (2000). Investigación de Operaciones en la Ciencia Administrativa. Pearson
- García et al. (2013). Simulación y Análisis de Sistemas con ProModel. Editorial Pearson.
- Srinivasan, G. (2010). Quantitative Models in Operations and Supply Chain Management. PHI Learning Private Limited
- Rardin, R. (2017). Optimization in Operations Research. Pearson
- Carter, M. et al. (2019). Operations Research A Practical Introduction. Taylor & Francis Group
- Aoroto Álvares, C., [et al] (2014) Operations research in business administration and managenent. Valencia: Universitat Politècnica de València
- Ravi Ravindran, A. (2008) Operations Research & Management Science Handbook. Taylor & Francis Group
- Rees, M. (2015). Business Risk and Simulation Modeling in Practice. John Wiley & Sons Ltd
- Sterman, J. (2000). Business Dynamics – Systems Thinking and Modeling for a Complex World. McGraw-Hill
- Winston, W. (2017) Microsoft Excel 2016 – Data Analysis and Busines Modeling. Microsoft press
- Schaffernicht, M. (2006). *Dinámica de Sistemas – Tomo 1: Fundamentos*.
- Alvarez, H. (2011). Introducción a la Simulación. Universidad Tecnológica de Panamá
- Cassandras & Lafortune. (2008). *Introduction to Discrete Event Systems*. Springer
- Bandyopadhyay, S. et al (2014). Discrete and Continuous Simulation: Theory and Practice. Taylor & Francis Group
- García et al. (2013). *Simulación y Análisis de Sistemas con ProModel*. Editorial Pearson.
- Schroeder et al. (2011). *Administración de Operaciones*. McGraw-Hill
- Render, B. & Heizer, J. (2014). *Principios de Administración de Operaciones*. Pearson
- Chase, R. & Jacobs, F. (2014). *Administración de Operaciones, Producción y Cadena de Suministro.* McGraw – Hill
- Rees, M. (2015). *Business Risk and Simulation Modeling in Practice*. John Wiley & Sons Ltd
- Sterman, J. (2000). *Business Dynamics – Systems Thinking and Modeling for a Complex World*. McGraw-Hill
- Winston, W. (2017) *Microsoft Excel 2016 – Data Analysis and Busines Modeling*. Microsoft press
- Slack, N., et al. (2016) . *Operations Management*. Pearson
- Stevenson, W. (2015*). Operations Management*. McGraw-Hill

Contacto

Ricardo Caballero, M.Sc.
Docente Tiempo Completo
Facultad de Ingeniería Industrial
Universidad Tecnológica de Panamá | Centro Regional de Chiriquí

E-Mail: ricardo.caballero@utp.ac.pa
Social: LinkedIn | ResearchGate
Website: https://www.academia.utp.ac.pa/ricardo-caballero

Project Manager
Giii Grupo de Investigación en Ingeniería Industrial
Website: www.giii.utp.ac.pa