**Distribución Uniforme**

**Teoría**

La Distribución Uniforme se caracteriza por ser “plana” y asimilarse a una función constante en un intervalo cerrado , los cuales son, a la vez, los dos parámetros que la distribución recibe. Su función de densidad de probabilidad está dada por:

La media de esta distribución viene dada por:

Y su varianza por:

Información obtenida de la Presentación 9 de clase del curso Probabilidad y Estadística.

**Gráfica en R (Script)**

Para graficar la función de densidad de la distribución uniforme se ha utilizado la función *curve()* que viene preinstalada en el paquete para el trazado de funciones de R. Primero, se deben ingresar los valores alfa y beta en las variables *alfaU* y *betaU* que, a modo de ejemplo, se le han asignado los valores 0 y 4, respectivamente. El usuario puede con toda libertad cambiar estos valores introduciéndolos en lugar de los dígitos preestablecidos.

*alfaU 0*

*betaU 4*

Posteriormente, se pasa a la creación del gráfico con la función *curve()*. Esta, recibirá 14 parámetros, de los cuales los primeros cuatro hacen referencia al trazado de la gráfica y, los demás, a la estética de la gráfica. El primer parámetro debe ser una expresión que haga referencia a lo que se graficará con respecto a una variable independiente X:

*dunif(x, min = alfaU, max = betaU)*

Esta función hace referencia a la función de densidad de la Distribución Uniforme, ya que la palabra reservada para dicha distribución “unif”, va antecedida de una letra “d”, densidad, definiendo el mínimo y el máximo correspondiendo a las variables alfaU y betaU respectivamente. Luego, se delimitan los límites de la gráfica en los parámetros *from* y *to*, de dónde a dónde se graficará, seguido del número de particiones (*n)* que se hará entre dichos límites.

*from = (alfaU - 3)*

*to = (betaU + 3)*

*n = 100000*

Además, para evitar confusiones, se le agrega el parámetro *add = FALSE*, ya que, este, asegurará que se dibuje la función en un nuevo gráfico evitando así que se dibuje uno sobre otro. Luego, se agrega el título en el parámetro *main* utilizando la función *paste*(), la cual concatena textos y caracteres para hacer más específico el mismo, en el ejemplo se agrega el carácter de escape \ seguido de una n para señalar un quiebre de línea y se utiliza el parámetro *sep* para definir el separador entre el texto o los valores pasados. Posteriormente, se identifican los ejes en los parámetros *xlab* y *ylab*.

*main = paste("DISTRIBUCION UNIFORME \n CON ALFA =",*

*alfaU, "Y BETA =", betaU, sep = " ")*

*xlab = "x", ylab = "f(x)"*

Asimismo, se define la estética de la gráfica empezando por el color en el parámetro *col* el cual acepta colores con su nombre en inglés o también sus identificaciones hexadecimales. Seguido del tipo de línea, en este caso sólido, correspondiendo al valor 1 asignado en el parámetro *lty* y el grosor de esta con *lwd*, entre más grande sea el valor, más gruesa será la línea. Además, se fija el estilo de fuente del texto y de los ejes con *font* y *font.lab* respectivamente con el valor de 2, equivalente a “bold”.

*col = "blue",*

*lty = 1, lwd = 3,*

*font.lab = 2,*

*font = 2*

Por otra parte, se le agrega una línea horizontal con la función siendo el parámetro *h*, el valor del intercepto con el eje Y:

*abline(h = (1/(betaU - alfaU)), lty = 2, col = "blue", lwd = 2)*

**Distribución Triangular**

**Teoría**

La Distribución Triangular se caracteriza por formar un triángulo con pico en el valor de la moda. Esta distribución recibe tres parámetros: un valor mínimo (a), un valor máximo (b) y el valor de la moda; correspondiendo los tres al dominio de la variable independiente X de la función de densidad, que viene, según información proveída por la Xunta de Galicia, por:

La media de esta distribución viene dada por:

Y su varianza por:

**Gráfica en R (Script):**

Para graficar la función de densidad de la Distribución Triangular se necesita instalar y cargar el paquete ggplot2:

*install.packages(‘ggplot2’)*

*library(ggplot2)*

Posteriormente, se introducen los valores requeridos en las variables *aMinimT*, *bMaximT*, *modaT* definiendo, a modo de ejemplo, los valores en 2, 9 y 3.5 respectivamente. Estos pueden ser modificados por el usuario introduciendo el nuevo valor en lugar de los dígitos establecidos.

*aMinimT 2*

*bMaximT 9*

*modaT 3.5*

Una vez se tienen los parámetros, se establece un dominio para la variable aleatoria continua X de la distribución. Por medio de una secuencia definimos el dominio de X, esta función necesita un límite inferior, un límite superior y una cantidad de particiones.

*seq(from = ‘limite\_inferior’, to = ‘limite\_superior’, length.out = ‘num\_particiones’)*

Se hace este procedimiento cuatro veces, ya que se necesita un dominio para la primera parte de la función compuesta que va desde el valor mínimo “a” hasta el valor de la moda, otro para el tramo delimitado por el valor de la moda y el valor del máximo “b” y, los otros dos, definidos afuera de los límites de la función de densidad con valor igual a 0.

*xValuesT1 <- seq(from = aMinimT, to = modaT, length.out = 1000)*

*xValuesT2 <- seq(from = modaT, to = bMaximT, length.out = 1000)*

*xValuesT3 <- seq(from = (aMinimT - 1), to = aMinimT, length.out = 1000)*

*xValuesT4 <- seq(from = bMaximT, to = (bMaximT + 1), length.out = 1000)*

Luego, se obtienes los valores de probalidad para cada X dentro de los vectores de datos. Cabe recalcar que R no tiene ninguna palabra reservada destinada para la Distribución Triangular por lo que, se ha ingresado a mano para definir dichos valores. Destinando *yTriang1* para la primera función dentro de la función compuesta y *yTriang2* para la segunda.

*yTriang1 = (2)\*(„Dominio\_1\_de\_X” - aMinimT)/((bMaximT - aMinimT)\*(modaT - aMinimT))*

*yTriang2 = (2)\*(bMaximT –„ Dominio\_3\_de\_X“)/((bMaximT - aMinimT)\*(bMaximT - modaT))*

Esto evalúa cada valor de X dentro de cada vector de datos en cada expresión y registra otro vector con los datos correspondientes para cada caso. En el caso de los valores de X que están fuera de los límites de la función, se multiplica todo el vector de datos por 0 para asignarle el valor a la probabilidad. Esto, se hace dentro del dataset creado con la función *data.frame*(), ya que GGPLOT2 solo acepta datasets como data.

datosT1 <- data.frame(

xTriang1 = xValuesT1,

yTriang1 = (2)\*(xValuesT1 - aMinimT)/((bMaximT - aMinimT)\*(modaT - aMinimT))

)

datosT2 <- data.frame(

xTriang2 = xValuesT2,

yTriang2 = (2)\*(bMaximT - xValuesT2)/((bMaximT - aMinimT)\*(bMaximT - modaT))

)

datosT3 <- data.frame(

xTriang3 = xValuesT3,

yTriang3 = xValuesT3\*0

)

datosT4 <- data.frame(

xTriang4 = xValuesT4,

yTriang4 = xValuesT4\*0

)

Para la creación del gráfico se colocan los datos en la función *ggplot*() y se agrega un tipo de gráfico, en este caso, *geom\_line*(). Para agregar los datos en *ggplot*() primero se coloca el *data.frame*() y, como tenemos cuatro dominios distintos, le pasaremos el que corresponde a la primera parte de la función de densidad que es *datosT1* y, con la función *aes*() se especifican los ejes.

ggplot(datosT1, aes(x = xTriang1, y = yTriang1)) + geom\_line()

En los parámetros de la función *geom\_line*() se agrega *colour* y *size* para darle estilo a la línea a graficar.

geom\_line(size = 1.5, colour = "blue")

Luego, agregamos otras tres funciones *geom\_line*() para graficar los otros tres dominios correspondientes. Las tres llevarán en sus primeros dos parámetros *size* y *colour* para definir su estilo antes de ser graficado, lo cual también se puede hacer. Seguidamente, se colocará en el parámetro *data* los *data.frames*() restantes en cada función *geom\_line*() respectivamente y siempre con la función *aes*() para definir los ejes; esto, con el objetivo de expandir los límites de la gráfica y trazar las líneas con sus valores correspondientes.

geom\_line(size = 1.5, colour = "blue", data = datosT2,

aes(x = xTriang2, y = yTriang2, colour = "blue")) +

geom\_line(size = 1.5, colour = "blue", data = datosT3,

aes(x = xTriang3, y = yTriang3, colour = "blue")) +

geom\_line(size = 1.5, colour = "blue", data = datosT4,

aes(x = xTriang4, y = yTriang4, colour = "blue"))

Posteriormente, se utilizará la función *geom\_vline*() para dibujar una línea vertical con el intercepto en el valor de X correspondiente a la moda e indicar, a su vez, el valor máximo de la Distribución Triangular.

geom\_vline(xintercept = modaT, lty = 2, colour = "blue", size = 1)

Además, se agregará una línea horizontal con el valor del intercepto en y = 0 para delimitar el espacio muestral de la Distribución Triangular de Probabilidad, con la función *geom\_hline*().

geom\_hline(yintercept = 0, lty = 1, colour = "black")

A continuación, se le agregará la función *labs*() con la cual asignaremos un título al gráfico y nombraremos sus ejes. Con el parámetro *x* se nombra el eje x, se agrega el carácter de escape \ seguido de una letra n “\n” para indicar un quiebre de línea antes del nombre y así despegarlo un poco del eje. Asimismo, con el parámetro *y* se describe el título del eje Y. Por último, se utiliza el parámetro *title* para agregarle un título al gráfico y, junto con la función *paste*() poder concatenar letras y números y así poder especificar tanto el nombre como los parámetros con los que la función se ha graficado, si se gusta.

x = "\n x"

y = "f(x) "

title = paste("GRAFICA DE DIST. TRIANGULAR \n CON MINIMO (a) =",

aMinimT, "| MAXIMO (b) =", bMaximT, "| MODA =",

modaT, sep = " ")

Por último, se le agrega la función *theme*() para darle estilos generales a la gráfica. Con *plot.title* *= element\_text*() formatearemos el texto correspondiente al título; con *hjust = 0.5* se justifica al centro el texto y, con *size = rel(1.2),* se agranda un poco el mismo. Además, agregando el parámetro *legend.position = “none”* aseguraremos que no hayan leyendas al lado del gráfico; esto se puede omitir si se gusta, pero se recomienda que, en caso de omitirlo y agregar leyenda, se formatee para mejorar su estética.

theme(plot.title = element\_text(hjust = 0.5, size = rel(1.2)),

legend.position = "none")

**Distribución Normal**

**Teoría**

La Distribución Normal es considerada la distribución de probabilidad continua más importante en todo el campo estadístico, ya que su figura, denominada curva normal, describe de manera aproximada muchos fenómenos del día a día. Esta distribución tiene una variable independiente X y recibe dos parámetros, la media () y la desviación estándar (). Su función de densidad está denotada por:

Puesto que la función de densidad no puede integrarse fácilmente, se suele estandarizar las distribuciones a una “Distribución Normal Estándar” con y , con el objetivo de utilizar tablas especiales con los valores ya obtenidos para algunos valores normalizados. Llegando a una función de densidad de una Distribución Normal más simple cuando ya está estandarizada:

Y, a través de la siguiente ecuación se puede estandarizar cualquier modelo de Distribución Normal y poder utilizar los valores de las tablas:

Información obtenida de la Presentación 9 de clase del curso Probabilidad y Estadística.

**Gráfica en R (Script):**

Para el trazado de la función de densidad de Distribución Normal se utilizará la función preinstalada *curve*() por lo que, no hace falta instalar ni cargar nada más. Esta distribución recibe dos parámetros media () y la desviación estándar () que se alojan en las variables *mediaNorm* y *desvestNorm* que, a modo de ejemplo, se han igualado a 3 y 0.5 respectivamente. El usuario puede cambiar los valores libremente ingresando los nuevos valores en vez de los dígitos ya insertados.

mediaNorm <- 3

desvestNorm <- 0.5

Para graficar la función se manda a llamar a la función *curve*() y se le pasarán una serie de parámetros. Primero, se le pasará una expresión que, en este caso, es la palabra reservada de R que regresa la función de densidad de probabilidad normal (cabe recalcar que no hace falta normalizar los valores de X ni tampoco que la media sea igual a cero y la desviación estándar igual a uno):

*dnorm(valoresX, mean = “media”, sd = “Desviación\_Estándar”)*

*dnorm(x, mean = mediaNorm, sd = desvestNorm)*

En el caso del ejemplo, en el parámetro *valoresX,* se pasa únicamente el carácter “x”, ya que el primer parámetro de la función *curve*() exige que se le pase en la primera expresión una variable independiente X que se simultaneara entre los límites de graficación que se indiquen con los parámetros *from* y *to*. Sin embargo, la función se puede graficar entre muchos rangos dependiendo del modelo de distribución al que se quiera aplicar (si es estándar se graficará de -∞ a +∞, por ejemplo).

Para solventar este problema, si ya se sabe que la media más menos tres veces la desviación estándar abarca aproximadamente el 97.7% de los datos (). Entonces la función se graficará desde hasta :

from = mediaNorm - (3 \* desvestNorm)

to = mediaNorm + (3 \* desvestNorm)

Además, entre estos límites se harán 10,000 particiones para evaluar la variable independiente X, definido por el parámetro *n = 10000*. Seguidamente, se le dará título a los ejes utilizando los parámetros *xlab* y *ylab*. Y, con el parámetro *add = FALSE*, se garantiza que no hayan confusiones al forzar a que se cree un nuevo plano cartesiano para trazar la función.

ylab = "f(x)", xlab = "\n x"

Además, se le agregará un título general a la gráfica con el parámetro *main* y, con ayuda de la función *paste*(), se concatenarán una serie de caracteres junto con los valores de la media y de la desviación estándar que se le pasaron a la función para identificar y distinguir la misma. Utilizando, también, el parámetro *sep = “ “* para hacer que la función separe cada elemento por un espacio.

main = paste("GRAFICA DE DIST. NORMAL \n CON MEDIA =",

mediaNorm, "Y DESVEST =", desvestNorm, sep = " ")

Luego, se define el estilo de la gráfica empezando por el tipo de línea *lty* *= 1,* que describe una línea sólida continua, y *lwd = 3*, que define el grosor de la misma. Seguidamente, con *font.lab* y *font*, se define el estilo de fuente y, en este caso, igualándolas a 2 serán “bold”. Finalizando por definir el color de la línea con el parámetro *col*.

Por último, se agrega una línea vertical con la función *abline*(), ya que, en muchas ocasiones, es muy útil identificar cuál es el centro de la distribución que equivale a la media de la misma. Esta, con el parámetro *v = mediaNorm* se garantiza que sea vertical y que su intercepto sea en la media de la distribución.

abline(v = mediaNorm, col = "blue", lty = 2, lwd = 2)

**Referencias Biblográficas**

Xunta de Galicia. (octubre del 2014). Distribuciones de Probabilidad. 68-72. Recuperado de https://www.sergas.es/Saude-publica/Documents/1899/Ayuda\_Epidat\_4\_Distribuciones\_de\_probabilidad\_Octubre2014.pdf