

# UNIDAD 3: Práctica 15 - Distribuciones de probabilidad continuas. Usando la interfaz gráfica (R-Commander)

Caterine Melissa Guerrero España

2022-09-05

## 1. CÁLCULO DE PROBABILIDADES

**Ejemplo 1:** Una persona informal hace esperar a su pareja aleatoriamente entre 0 y 90 minutos. Harto de esta situación, la persona que sufre la espera se plantea un ultimátum; si al día siguiente su pareja tarda menos de 15 minutos mantiene la relación, si la espera está entre 15 y 55 minutos, decide en la siguiente cita con los mismos criterios, mientras que si tarda más de 55 minutos la relación termina en ese momento.

a) Calcule la probabilidad de que la relación continúe hasta la siguiente cita.

Para que la relación se mantenga hasta la próxima cita, es porque la persona ha esperado a su pareja menos de 55 minutos (a lo sumo 55 minutos), por lo que debemos calcular la probabilidad acumulada en una distribución uniforme de que la variable tome el valor de 55 (el área entre 0 y 55). El procedimiento para encontrar distribuciones acumuladas de una uniforme es el siguiente; en el Menú Distribuciones seleccionamos la opción Distribuciones continuas, luego seleccionamos Distribución uniforme y finalmente la opción Probabilidades uniforme...

```
library(Rcmdr)

## Warning: package 'Rcmdr' was built under R version 4.1.3
## Loading required package: splines
## Loading required package: RcmdrMisc
## Warning: package 'RcmdrMisc' was built under R version 4.1.3
## Loading required package: car
## Warning: package 'car' was built under R version 4.1.3
## Loading required package: carData
## Warning: package 'carData' was built under R version 4.1.3
## Loading required package: sandwich
## Warning: package 'sandwich' was built under R version 4.1.3
## Loading required package: effects
## Warning: package 'effects' was built under R version 4.1.3
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
## La interfaz R-Commander sólo funciona en sesiones interactivas
##
## Attaching package: 'Rcmdr'
```

```
## The following object is masked from 'package:base':
##
##      errorCondition
```

```
library(car)
library(RcmdrMisc)

# Uniform Cumulative Probabilities
punif(c(55), min=0, max=90, lower.tail=TRUE)
```

```
## [1] 0.6111111
```

**Ejemplo 2:** Una empresa está buscando personal para su departamento de mercadeo. El perfil solicitado es el de sujetos extrovertidos y creativos. Se han presentado 50 candidatos y la empresa ha establecido como criterio de selección que los candidatos superen el percentil 80 en creatividad y extroversión. Sabiendo que la variable extroversión (X) se distribuye según una Normal de media 5 y desviación típica 1, que la variable creatividad (Y) sigue una t-Student de 10 grados de libertad y que las puntuaciones de creatividad y extroversión son independientes entre si:

a) ¿Qué puntuaciones debe superar un aspirante en creatividad y extroversión para ser admitido?

Según el criterio de selección se debe superar el percentil 80, en ambas variables, para ser admitido. Se calculará pues el percentil 80 de la variable X e Y, utilizando los cuantiles-normales para la variable X:

Para obtener los cuantiles (valores que dejan por encima o por debajo un área específica) de cualquier distribución continua, en especial los de la distribución normal el procedimiento es el siguiente. En el Menú Distribuciones seleccionar la opción Distribuciones continuas, posteriormente Distribución normal y finalmente la opción Cuantiles normales

```
# Normal Quantiles
qnorm(c(0.8), mean=5, sd=1, lower.tail=TRUE)
```

```
## [1] 5.841621
```

Para obtener los cuantiles de la distribución t de Student el procedimiento similar, solamente aplicado a dicha distribución. En el cuadro que se mostrará (llamado Cuantiles t) la única diferencia con la distribución normal, radica en que aquí en lugar de especificar el valor de la media y la desviación típica se especifica el número de grados de libertad, los demás datos se llenan con los mismos criterios.

```
# t Quantiles
qt(c(0.8), df=10, lower.tail=TRUE)
```

```
## [1] 0.8790578
```

c) Si se extraen al azar 16 candidatos, ¿cuál es la probabilidad de que su media aritmética en extroversión sea mayor que 4.5?

Se sabe que al extraer una muestra de una población normal de tamaño n, la media muestral, sigue otra distribución normal de media igual al de la poblacional y desviación típica  $\frac{\sigma}{\sqrt{n}}$

Para obtener dicha probabilidad en la Distribución normal en lugar de seleccionar la opción de cuantiles se selecciona Probabilidad binomiales (nos da la probabilidad acumulada de la variable); obteniendo el siguiente cuadro de dialogo, en el solamente debe especificar el valor de la media y de la desviación típica (debe escribirse el valor calculado de  $\frac{\sigma}{\sqrt{n}}$ ), y finalmente el valor a partir del cual encontrará la probabilidad acumulada (4.5 en nuestro caso), como se desea la probabilidad de observar datos mayores se elige la opción Cola derecha.

```
# Normal Cumulative Probabilities
pnorm(c(4.5), mean=5, sd=0.25, lower.tail=FALSE)
```

```
## [1] 0.9772499
```

Del mismo modo puede obtenerse la probabilidades acumuladas o los cuantiles para cualquier distribución continua (eligiendo la distribución adecuada).

## 2. GENERACIÓN DE MUESTRAS ALEATORIAS DE LAS DISTRIBUCIONES

**Ejemplo 1:** Generar 100 números aleatorios de una distribución Uniforme en el intervalo  $[-2, 4]$

El procedimiento para generar muestras aleatorias de una distribución uniforme es el siguiente: en el Menú Distribuciones se selecciona Distribuciones continuas, luego se elige Distribución uniforme y finalmente la opción Muestra de una distribución uniforme.

```
# Muestra de una distribución uniforme
UniformSamples <- as.data.frame(matrix(runif(1*100, min=-2, max=4), ncol=100))
rownames(UniformSamples) <- "sample"
colnames(UniformSamples) <- paste("obs", 1:100, sep="")
```

Para generar muestras de cualquier distribución el procedimiento es el mismo, teniendo en cuenta únicamente los parámetros que definen a cada una de las distribuciones.

- **Ejercicio 1:** Generar una muestra de tamaño  $n=200$  perteneciente a una población normal  $N(10; 2^2)$ .

```
# Muestra de una distribución normal
NormalSamples <- as.data.frame(matrix(rnorm(1*200, mean=10, sd=2), ncol=200))
rownames(NormalSamples) <- "sample"
colnames(NormalSamples) <- paste("obs", 1:200, sep="")
NormalSamples <- within(NormalSamples, {
  mean <- rowMeans(NormalSamples[,1:200])
})
```

- **Ejercicio 2:** ¿Cuál es la probabilidad a la derecha de 18.55 para una Variable aleatoria  $X$  con distribución Chi-cuadrado de 12 grados de libertad?

```
# Chi-Squared Cumulative Probabilities
pchisq(c(18.55), df=12, lower.tail=FALSE)
```

```
## [1] 0.09998251
```

- **Ejercicio 3:** Generar 100 números aleatorios de una distribución Normal con media 4.5 y desviación estándar 0.75

```
AleatorioNorm <- as.data.frame(matrix(rnorm(1*100, mean=4.5, sd=0.75),
  ncol=100))
rownames(AleatorioNorm) <- "sample"
colnames(AleatorioNorm) <- paste("obs", 1:100, sep="")
AleatorioNorm
```

```
##      obs1      obs2      obs3      obs4      obs5      obs6      obs7      obs8
## sample 4.219813 4.222492 5.145089 3.804264 5.206659 4.820586 3.458574 3.802672
##      obs9      obs10     obs11     obs12     obs13     obs14     obs15     obs16
## sample 5.112632 4.179419 4.381305 3.968996 6.163407 6.233573 4.212592 3.450223
##      obs17     obs18     obs19     obs20     obs21     obs22     obs23     obs24
## sample 5.443586 6.228501 3.633253 4.117993 3.576814 4.621358 5.507989 4.489265
##      obs25     obs26     obs27     obs28     obs29     obs30     obs31     obs32
## sample 5.517368 3.998447 7.364511 5.598441 5.027117 5.326559 5.370055 3.435352
##      obs33     obs34     obs35     obs36     obs37     obs38     obs39     obs40
## sample 4.176606 4.185291 3.817525 4.995592 4.899585 5.636669 6.069879 4.759084
##      obs41     obs42     obs43     obs44     obs45     obs46     obs47     obs48
## sample 5.357565 3.324745 4.343874 4.126898 3.853357 5.403742 5.184165 5.367037
##      obs49     obs50     obs51     obs52     obs53     obs54     obs55     obs56
## sample 3.932176 5.21855 3.022135 3.810804 3.981521 3.662812 4.370489 5.091018
```

```
##      obs57      obs58      obs59      obs60      obs61      obs62      obs63      obs64
## sample 3.885279 3.956076 4.065135 5.088447 4.881465 4.930958 3.889171 4.760031
##      obs65      obs66      obs67      obs68      obs69      obs70      obs71      obs72
## sample 5.092355 4.331496 4.607735 5.141139 4.524009 4.696006 4.477637 3.604263
##      obs73      obs74      obs75      obs76      obs77      obs78      obs79      obs80
## sample 4.292871 4.78879 5.436083 3.843037 5.056186 5.048245 4.381513 4.224395
##      obs81      obs82      obs83      obs84      obs85      obs86      obs87      obs88
## sample 4.508176 4.901966 4.03418 4.368144 4.767422 4.15732 3.515921 5.466819
##      obs89      obs90      obs91      obs92      obs93      obs94      obs95      obs96
## sample 3.418202 4.126709 5.033504 4.218361 4.99221 5.18517 4.251359 3.806907
##      obs97      obs98      obs99      obs100
## sample 5.421568 3.81332 5.994886 4.074674
```

- **Ejercicio 4:** Generar números aleatorios de una distribución exponencial, si la media es 2500

```
ExponentialSamples <- as.data.frame(matrix(rexp(1*100, rate=2500),
ncol=100))
rownames(ExponentialSamples) <- "sample"
colnames(ExponentialSamples) <- paste("obs", 1:100, sep="")
ExponentialSamples
```

```
##      obs1      obs2      obs3      obs4      obs5
## sample 0.0001082108 0.000484139 4.024564e-05 2.612071e-05 0.0001533326
##      obs6      obs7      obs8      obs9      obs10
## sample 0.000338344 1.840417e-05 0.0004913319 0.0004186567 0.0005271743
##      obs11      obs12      obs13      obs14      obs15
## sample 0.000238994 0.0003455578 0.0006679912 0.0001599647 4.485413e-05
##      obs16      obs17      obs18      obs19      obs20
## sample 0.0003345518 0.0006850366 0.0006050807 0.0003074613 2.367149e-05
##      obs21      obs22      obs23      obs24      obs25
## sample 0.0003494124 0.000623509 0.0003003577 0.0007830807 1.06302e-05
##      obs26      obs27      obs28      obs29      obs30
## sample 0.0006469909 1.78122e-05 0.0003419719 0.0004985837 0.000500091
##      obs31      obs32      obs33      obs34      obs35
## sample 0.0005736348 0.0006176852 0.0001992404 8.173067e-05 0.001675255
##      obs36      obs37      obs38      obs39      obs40
## sample 0.0004557693 1.809964e-05 0.001057111 0.0007453802 1.472576e-05
##      obs41      obs42      obs43      obs44      obs45
## sample 2.280245e-05 7.491761e-06 0.000556968 0.0003600963 0.0008099822
##      obs46      obs47      obs48      obs49      obs50      obs51
## sample 0.001676311 0.000965169 3.6799e-05 0.000582029 0.0004128672 5.61331e-05
##      obs52      obs53      obs54      obs55      obs56
## sample 0.0008022012 0.000549188 0.0008671663 0.001291985 8.345737e-05
##      obs57      obs58      obs59      obs60      obs61
## sample 3.69705e-06 0.0001541223 0.0005676787 0.0007321537 0.0004037573
##      obs62      obs63      obs64      obs65      obs66
## sample 0.0001204559 0.0004310002 0.0001096008 1.339945e-05 0.0004533402
##      obs67      obs68      obs69      obs70      obs71
## sample 0.0009679905 0.0005725927 8.64104e-05 4.190643e-05 0.0006360929
##      obs72      obs73      obs74      obs75      obs76
## sample 9.290261e-05 0.0003744115 0.0001578641 0.0002887595 0.0002101792
##      obs77      obs78      obs79      obs80      obs81
## sample 0.0004138947 3.409626e-05 5.340474e-05 0.0002973159 0.0006795954
##      obs82      obs83      obs84      obs85      obs86
## sample 0.0002615741 0.0003060938 0.000396048 0.0001080604 0.0001957965
```

##		obs87	obs88	obs89	obs90	obs91
## sample	4.646867e-05	0.0001101401	0.0001443353	0.000838116	4.859306e-05	
##		obs92	obs93	obs94	obs95	obs96
## sample	2.638691e-05	7.876716e-06	0.0002130245	7.178211e-05	0.0002414894	
##		obs97	obs98	obs99	obs100	
## sample	3.452779e-05	0.0006937731	6.154803e-05	0.0003414515		