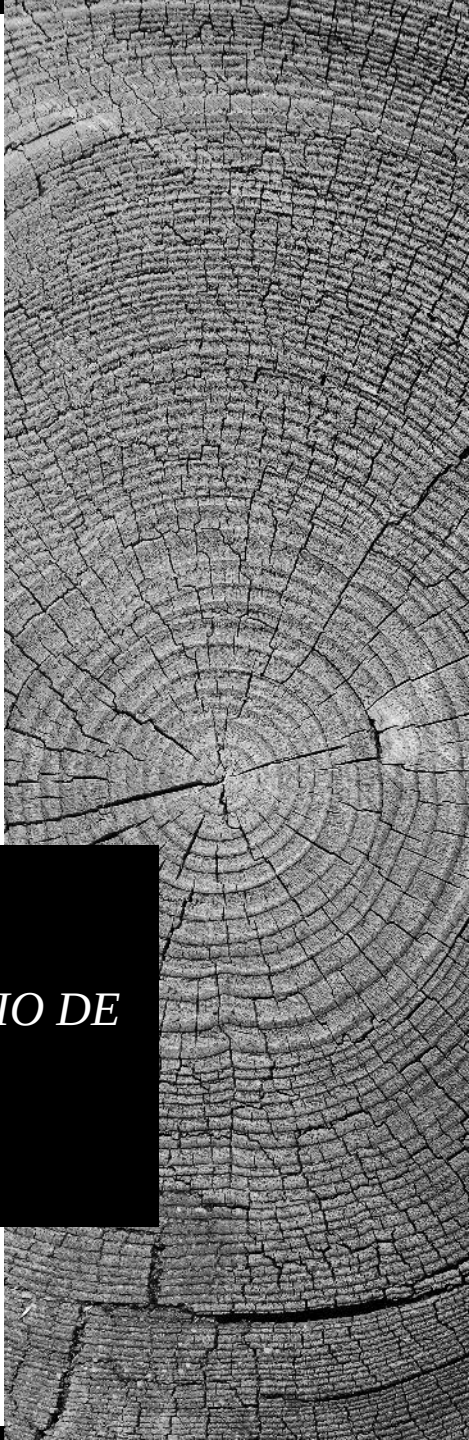
ANALISIS ESTADISTICO DE DATOS EN DATA FRAMES

WOODGROVE BANK

CURSO INTRODUCTORIO DE R



FILTRADO NATURAL EN DATA FRAMES

OPCIÓN BÁSICA

OPERADORES RELACIONALES Y LÓGICOS

Usado para incluir condiciones en los filtros

Relacional	Descripción
<	Menor que
>	Mayor que
==	Igual
<=	Menor o igual que
>=	Mayor o igual que
!=	Diferente de
%in%	Pertenece al conjunto
is.na	Es NA
!is.na	No es NA

Lógico	Descripción
&	boolean and
|	boolean o
xor	or inclusivo
!	no
any	cualquiera true
all	todos verdaderos

Operadores relacionales y lógicos. Fuente: Elaboración propia.

FUNCIONES BÁSICAS PARA FILTRAR EN R

HAY VARIAS, LAS CUALES SE RESUMEN A CONTINUACIÓN

Hay variables funciones que nos permiten filtrar data frames y son (las cuales se explicarán en las secciones siguientes):

1. Función *corchete*.
2. Función *subset*.
3. Función *sample*.
4. Función *filter* de la *base* de R.
5. Función *filter* del paquete *dplyr*.

LA FUNCIÓN CORCHETE

ES EL FILTRO MÁS BÁSICO DE R, SE BASA EN LA POSICION DE LOS OBJETOS EN EL DATA FRAME

Una opción es ejecutando *datosCompleto[i,j]*, donde *i* y *j* son las filas y columnas que se va a utilizar o quitar, respectivamente.

1. Si escribimos *datosCompleto[, j]* (sin información para la fila *i*), indicaremos que queremos construir un data frame que tenga en cuenta todas las filas (observaciones), pero solo la columna (variable) *j*.
2. Si escribimos *datosCompleto[i,]* (sin información para la columna *j*), indicaremos que queremos construir un data frame que tenga en cuenta solo la fila (observación) *i*, pero con todas las columnas (variables).
3. Las expresiones *-i*, *-j* (con el signo menos), indicarán quitar la fila *i* y/o la columna *j*.
4. Lo anterior también es válidos para vectores.

LA FUNCIÓN SUBSET

SUBCONJUNTO DE LA TABLA QUE CUMPLEN CIERTAS CONDICIONES

Arguments

`x`	object to be subsetted.
`subset`	logical expression indicating elements or rows to keep: missing values are taken as false.
`select`	expression, indicating columns to select from a data frame.
`drop`	passed on to `[` indexing operator.
...	further arguments to be passed to or from other methods.

FUNCIÓN SAMPLE

PARA TOMAR MUESTRAS ALEATORIAS SIMPLES DE UNA POBLACIÓN

La función *sample* se utiliza para tomar una muestra aleatoria de tamaño n de un conjunto de datos. El muestreo se puede hacer con reemplazo y sin reemplazo. Es importante resaltar que, cada vez que llamamos *sample*, se generan muestras aleatorias diferentes. Pero, si antes fijamos una semilla de R (con *set.seed*), con un número específico como argumento, obtendremos los mismos resultados cada vez que se ejecute el comando.

Arguments

`x`	either a vector of one or more elements from which to choose, or a positive integer.
`size`	a positive number, the number of items to choose from.
`replace`	a non-negative integer giving the number of items to choose.
`prob`	should sampling be with replacement?

FUNCIÓN FILTER

FILTROS DE FILAS CON CONDICIONES MÁS ESPECÍFICAS

Usage

```
filter(.data, ..., .preserve = FALSE)
```

Arguments

`.data`	A `data.frame`.
`...`	Logical predicated defined in terms of the variables in `.data`. Multiple conditions are combined with `&`. Arguments within `...` are automatically quoted and evaluated within the context of the `data.frame`.
`.preserve`	`logical(1)`. Relevant when the .data input is grouped. If `.preserve = FALSE` (the default), the grouping structure is recalculated based on the resulting data, otherwise the grouping is kept as is.

ANALISIS ESTADÍSTICO CON DATOS. PARTE 1

ESTADISTICA DESCRIPTIVA ´GRÁFICA

LA FUNCIÓN PLOT

ENTORNO GENERAL PARA GRÁFICAS EN R

En R, la función `plot()` es usada de manera general para crear gráficos.

Esta función tiene un comportamiento especial, pues dependiendo del tipo de dato que le demos como argumento, generará diferentes tipos de gráfica. Además, para cada tipo de gráfico, podremos ajustar diferentes parámetros que controlan su aspecto, dentro de esta misma función.

Puedes imaginar a `plot()` como una especie de navaja Suiza multifuncional, con una herramienta para cada ocasión.

`plot()` siempre pide un argumento `x`, que corresponde al **eje X** de una gráfica. `x` requiere un vector y si no especificamos este argumento, obtendremos un error y no se creará una gráfica.

El resto de los argumentos de `plot()` son opcionales, pero el más importante es `y`. Este argumento también requiere un vector y corresponde al **eje Y** de nuestra gráfica.

FUNCIÓN PLOT

TIPOS DE GRÁFICAS

x	y	Gráfico
Continuo	Continuo	Diagrama de dispersión (*Scatterplot*)
Continuo	Discreto	Diagrama de dispersión, `y` coercionada a numérica
Continuo	Ninguno	Diagrama de dispersión, por número de renglón
Discreto	Continuo	Diagrama de caja (*Box plot*)
Discreto	Discreto	Gráfico de mosaico (Diagrama de Kinneman)
Discreto	Ninguno	Gráfica de barras
Ninguno	Cualquiera	Error

HISTOGRAMAS

PARA DATOS DE TIPO NUMÉRICO

Un histograma es una gráfica que nos permite observar la distribución de datos numéricos usando barras. Cada barra representa el número de veces (frecuencia) que se observaron datos en un rango determinado.

Para crear un histograma usamos la función `hist()`, que siempre nos pide como argumento `x` un vector numérico. El resto de los argumentos de esta función son opcionales. Si damos un vector no numérico, se nos devolverá un error.

BOXPLOT

ANALISIS DE DISTRIBUCIÓN DE DATOS Y DETREMINACIÓN DE OUTLIERS

Los diagrams de caja, también conocidos como de caja y bigotes son gráficos que muestra la distribución de una variable usando cuartiles, de modo que de manera visual podemos inferir algunas cosas sobre su dispersión, ubicación y simetría.

Una gráfica de este tipo dibuja un rectángulo cruzado por una línea recta horizontal. Esta linea recta representa la mediana, el segundo cuartil, su base representa el pimer cuartil y su parte superior el tercer cuartil. Al rango entre el primer y tercer cuartil se le conoce como intercuartílico (RIC). Esta es la caja.

Además, de la caja salen dos líneas. Una que llega hasta el mínimo valor de los datos en la variable o hasta el primer cuartil menos hast 1.5 veces el RIC; y otra que llegar hasta el valor máximo de los datos o el tercer cuartil más hasta 1.5 veces el RIC. Estos son los bigotes.

Usamos la función `plot()` para crear este tipo de gráfico, dando como argumento `x` un vector de factor o cadena de texto, y como argumento `y` un vector numérico.

DIAGRAMA DE DISPERSIÓN (SCATTERPLOT)

ENTRE DOS VARIABLES NUMERICAS CONTINUAS

Este tipo de gráfico es usado para mostrar la relación entre dos variables numéricas continuas, usando puntos. Cada punto representa la intersección entre los valores de ambas variables.

Para generar un diagrama de dispersión, damos vectores numéricos como argumentos `x` y `y` a la función `plot()`.

Veamos la relación entre las variables **age** y **balance** de `banco`.

```
plot(x = banco$age, y = banco$balance)
```

DIAGRAMA DE BARRAS Y DE TORTA

PARA DATOS DE TIPO CATEGÓRICO

Este es quizás el tipo de gráfico mejor conocido de todos. Una gráfica de este tipo nos muestra la frecuencia con la que se han observado los datos de una variable discreta, con una barra para cada categoría de esta variable.

La función `plot()` puede generar gráficos de barra si damos como argumento `x` un vector de factor o cadena de texto, sin dar un argumento `y` .

Por ejemplo, creamos una gráfica de barras de la variable educación (“education”) de `banco`

```
plot(x = banco$education)
```