# # Factores

{#factors}

En R, los factores son un tipo de datos que permiten categorías ordenadas con un conjunto fijo de valores aceptables.

Normalmente, se convierte una columna de tipo numérico o de caracteres en un factor si se desea establecer un orden intrínseco a los valores ("niveles") para que puedan mostrarse de forma no alfabética en gráficos y tablas. Otro uso común de los factores es normalizar las leyendas de los gráficos para que no fluctúen si ciertos valores están temporalmente ausentes de los datos.

En esta página se muestra el uso de las funciones del paquete **forcats** (nombre abreviado de "**Para** variables categóricas") y algunas funciones **básicas de** R. También se aborda el uso de **lubridate** y **aweek** para casos de factores especiales relacionados con semanas epidemiológicas.

Puedes encontrar una lista completa de las funciones de **los forcats** en línea [aquí](https://forcats.tidyverse.org/reference/index.html). A continuación mostramos algunas de las más comunes.

## Preparación

### Cargar paquetes

Este trozo de código muestra la carga de los paquetes necesarios para los análisis. En este manual destacamos p\_load() de **pacman**, que instala el paquete si es necesario y lo carga para su uso. También puede cargar los paquetes instalados con library() de **.** Consulta la página sobre [los fundamentos de R](#r-basics) para obtener más información sobre los paquetes de R.

### Importar datos

Importamos los datos de casos de una epidemia de ébola simulada. Si quieres seguir el proceso, [clica para descargar linelist "limpia"](https://github.com/epirhandbook/Epi_R_handbook/raw/master/data/case_linelists/linelist_cleaned.rds) (como archivo .rds). Importe sus datos con la función import() del paquete **rio** (acepta muchos tipos de archivos como .xlsx, .rds, .csv - vea la página de [importación y exportación](#import-and-export) para más detalles).

### Nueva variable categórica

Para la demostración en esta página utilizaremos un escenario común - la creación de una nueva variable categórica.

Ten en cuenta que si convierte una columna numérica en una de tipo factor, no podrás calcular estadísticas numéricas sobre ella.

#### Crear columna

Utilizamos la columna existente days\_onset\_hosp (días desde el inicio de los síntomas hasta el ingreso en el hospital) y creamos una nueva columna delay\_cat clasificando cada fila en una de varias categorías. Lo hacemos con la función **dplyr** case\_when(), que aplica secuencialmente criterios lógicos (lado derecho) a cada fila y devuelve el valor correspondiente del lado izquierdo para la nueva columna delay\_cat. Lea más sobre case\_when() en [Limpieza de datos y funciones básicas](#cleaning-data-and-core-functions).

#### Orden de valores por defecto

Tal y como se creó con case\_when(), la nueva columna delay\_cat es una columna categórica del tipo Character - aún no es un factor. Así, en una tabla de frecuencia, vemos que los valores únicos aparecen en un orden alfanumérico por defecto - un orden que no tiene mucho sentido intuitivo:

Del mismo modo, si hacemos un gráfico de barras, los valores también aparecen en este orden en el eje x (ver la página de [fundamentos de ggplot](#ggplot-basics) para más información sobre **ggplot2** - el paquete de visualización más común en R).

## Convertir en factor

Para convertir una columna numérica o de caracteres a de tipo factor, puedes utilizar cualquier función del paquete **forcats** (muchas se detallan [a continuación](#fct_adjust)). Las convertirán a de tipo factor y luego también realizarán o permitirán cierto ordenamiento de los niveles - por ejemplo usando fct\_relevel() le permite especificar manualmente el orden de los niveles. La función as\_factor() simplemente convierte el tipo sin ninguna otra capacidad.

La función **base de** R factor() convierte una columna en factor y permite especificar manualmente el orden de los niveles, como un vector de caracteres a su argumento niveles =.

A continuación utilizamos mutate() y fct\_relevel() para convertir la columna delay\_cat de tipo carácter a tipo factor. La columna delay\_cat se crea en la sección de [preparación](#fct_newcat) anterior.

Los "valores" únicos de esta columna se consideran ahora "niveles" del factor. Los niveles tienen un orden, que puede imprimirse con la función de levels(), o alternativamente verse en una tabla de recuento mediante table() de o tabyl() de **janitor**. Por defecto, el orden de los niveles será alfanumérico, como antes. Ten en cuenta que NA no es un nivel de factor.

La función fct\_relevel() tiene la utilidad adicional de permitirle especificar manualmente el orden de los niveles. Simplemente escriba los valores de nivel en orden, entre comillas, separados por comas, como se muestra a continuación. Ten en cuenta que la ortografía debe coincidir exactamente con los valores. Si desea crear niveles que no existen en los datos, utiliza [fct\_expanden su lugar](#fct_add)).

Ahora podemos ver que los niveles están ordenados, como se especificó en el comando anterior, en un orden sensato.

Ahora el orden de la trama también tiene un sentido más intuitivo.

## Añadir o quitar niveles

### Añadir

Si necesita añadir niveles a un factor, puede hacerlo con fct\_expand(). Basta con escribir el nombre de la columna seguido de los nuevos niveles (separados por comas). Al tabular los valores, podemos ver los nuevos niveles y los recuentos de cero. Puedes utilizar table() de , o tabyl() de **janitor**:

Nota: existe una función especial **de forcats** para añadir fácilmente valores faltantes (NA) como nivel. Véase la sección sobre valores [ausentes](#fct_missing) más adelante.

### Gota

Si utiliza fct\_drop(), los niveles "no utilizados" con recuento cero se eliminarán del conjunto de niveles. Los niveles que hemos añadido anteriormente ("No admitido en un hospital") existen como nivel, pero ninguna fila tiene realmente esos valores. Por tanto, se eliminarán aplicando fct\_drop() a nuestra columna de factores:

## Ajustar el orden de los niveles

El paquete **forcats** ofrece funciones útiles para ajustar fácilmente el orden de los niveles de un factor (después de haber definido una columna como de tipo factor):

Estas funciones pueden aplicarse a una columna de factores en dos contextos:

1. A la columna del dataframe, como es habitual, para que la transformación esté disponible para cualquier uso posterior de los datos
2. Dentro de un gráfico, para que el cambio se aplique sólo dentro del gráfico

### Manualmente

Esta función se utiliza para ordenar manualmente los niveles de los factores. Si se utiliza en una columna no factorial, la columna se convertirá primero en de tipo factor.

Dentro del paréntesis, indique primero el nombre de la columna del factor y, a continuación, indique

* Todos los niveles en el orden deseado (como un vector de caracteres c()), o
* Un nivel y se corrige la colocación utilizando el argumento after =

He aquí un ejemplo de redefinición de la columna delay\_cat (que ya es el tipo Factor) y especificando todo el orden de niveles deseado.

Si sólo quiere mover un nivel, puede especificarlo a fct\_relevel() solo y dar un número al argumento after = para indicar en qué lugar del orden debe estar. Por ejemplo, el comando siguiente desplaza "<2 días" a la segunda posición:

### Dentro de un gráfico

Los comandos **forcats** pueden utilizarse para establecer el orden de los niveles en el dataframe, o sólo dentro de un gráfico. Al utilizar el comando para "envolver" el nombre de la columna dentro del comando de trazado ggplot(), puede invertir/nivelar/etc. la transformación sólo se aplicará dentro de ese trazado.

A continuación, se crean dos gráficos con ggplot() (véase la página de [fundamentos de ggplot](#ggplot-basics)). En el primero, la columna delay\_cat se asigna al eje x del gráfico, con su orden de nivel por defecto como en linelist de datos. En el segundo ejemplo se envuelve dentro de fct\_relevel() y se cambia el orden en el gráfico.

Ten en cuenta que el título del eje x por defecto es ahora bastante complicado - puede sobrescribir este título con el argumento **ggplot2** labs().

### Invertir

Es bastante común que se quiera invertir el orden de los niveles. Basta con envolver el factor con fct\_rev().

Ten en cuenta que si desea revertir sólo una leyenda del gráfico pero no los niveles reales del factor, puede hacerlo con guides() (ver [consejos de ggplot](#ggplot-tips)).

### Por frecuencia

Para ordenar por la frecuencia con que el valor aparece en los datos, utiliza fct\_infreq(). Cualquier valor que falte (NA) se incluirá automáticamente al final, a menos que se convierta en un nivel explícito (véase [esta sección](#fct_missing)). Puedes invertir el orden envolviendo más con fct\_rev().

Esta función puede utilizarse dentro de un ggplot(), como se muestra a continuación.

### Por apariencia

Utiliza fct\_inorder() para establecer el orden de los niveles para que coincida con el orden de aparición en los datos, empezando por la primera fila. Esto puede ser útil si primero organiza cuidadosamente() los datos en el dataframe, y luego utiliza esto para establecer el orden de los factores.

### Por estadística resumida de otra columna

Puedes utilizar fct\_reorder() para ordenar los niveles de una columna por una estadística de resumen de otra columna. Visualmente, esto puede dar lugar a gráficos agradables en los que las barras/puntos ascienden o descienden de forma constante a través del gráfico.

En los ejemplos siguientes, el eje x es delay\_cat, y el eje y es la columna numérica ct\_blood (valor de umbral de ciclo). Los gráficos de caja muestran la distribución del valor CT por grupo delay\_cat. Queremos ordenar los gráficos de caja en orden ascendente por el valor de TC mediano del grupo.

En el primer ejemplo de abajo, se utiliza el orden por defecto de los niveles alfa-numéricos. Se puede ver que las alturas de los gráficos de caja están mezcladas y no en ningún orden particular. En el segundo ejemplo, la columna delay\_cat (asignada al eje x) se ha envuelto en fct\_reorder(), la columna ct\_blood se da como segundo argumento, y la "mediana" se da como tercer argumento (también podría usar "max", "mean", "min", etc). Por lo tanto, el orden de los niveles de delay\_cat reflejará ahora los valores ascendentes de la mediana del CT de cada grupo de delay\_cat. Esto se refleja en el segundo gráfico: los gráficos de caja se han reordenado de forma ascendente. Nótese cómo NA (missing) aparecerá al final, a menos que se convierta en un nivel explícito.

Observe que en este ejemplo no se requieren pasos previos a la llamada a ggplot() - la agrupación y los cálculos se realizan internamente en el comando ggplot.

### Por valor "final"

Utiliza fct\_reorder2() para los gráficos de líneas agrupadas. Ordena los niveles (y, por tanto, la leyenda) para que se alineen con la ordenación vertical de las líneas en el "final" del gráfico. Técnicamente hablando, "ordena por los valores y asociados a los valores x más grandes".

Por ejemplo, si tiene líneas que muestran los recuentos de casos por hospital a lo largo del tiempo, puede aplicar fct\_reorder2() al argumento color = dentro de aes(), de forma que el orden vertical de los hospitales que aparecen en la leyenda se alinee con el orden de las líneas en el extremo terminal del gráfico. Lea más en la [documentación en línea](https://forcats.tidyverse.org/reference/fct_reorder.html).

## Valores faltantes

Si tiene valores NA en su columna de factores, puede convertirlos fácilmente a un nivel con nombre como "Missing" con fct\_explicit\_na(). Los valores NA se convierten por defecto en "(Missing)" al final del orden de los niveles. Puedes ajustar el nombre del nivel con el argumento na\_level =.

A continuación, esta operación se realiza en la columna delay\_cat y se imprime una tabla con tabyl() con NA convertido en "Missing delay".

## Combinar niveles

### Manualmente

Puedes ajustar las visualizaciones de los niveles manualmente con fct\_recode(). Es como la función **dplyr recode**() (véase [Limpieza de datos y funciones básicas](#cleaning-data-and-core-functions)), pero permite la creación de nuevos niveles de factores. Si utiliza la función simple recode() en un factor, los nuevos valores recodificados serán rechazados a menos que ya hayan sido establecidos como niveles permitidos.

Esta herramienta también puede utilizarse para "combinar" niveles, asignando a varios niveles el mismo valor recodificado. Sólo hay que tener cuidado de no perder información. Considere la posibilidad de realizar estos pasos de combinación en una nueva columna (sin sobreescribir la columna existente).

fct\_recode() tiene una sintaxis diferente a la de recode(). recode() utiliza OLD = NEW, mientras que fct\_recode() utiliza NEW = OLD.

Los niveles actuales de delay\_cat son:

Los nuevos niveles se crean utilizando la sintaxis fct\_recode(columna, "nuevo" = "viejo", "nuevo" = "viejo", "nuevo" = "viejo") y se imprimen:

Aquí se combinan manualmente con fct\_recode(). Obsérvese que no se produce ningún error en la creación de un nuevo nivel "Menos de 5 días".

### Reducir a "Otros"

Puedes utilizar fct\_other() para asignar manualmente niveles de factor a un nivel "Otro". A continuación, todos los niveles de la columna hospital, aparte de "Hospital del Puerto" y "Hospital Central", se combinan en "Otros". Puedes proporcionar un vector para mantener =, o eliminar =. Puedes cambiar la visualización del nivel "Otro" con other\_level =.

### Reducir por frecuencia

Puedes combinar los niveles del factor menos frecuente automáticamente utilizando fct\_lump().

Para "agrupar" muchos niveles de baja frecuencia en un grupo "Otros", haga una de las siguientes cosas:

* Establezca n = como el número de grupos que desea conservar. Los n niveles más frecuentes se mantendrán, y todos los demás se combinarán en "Otros".
* Establezca prop = como la proporción de frecuencia del umbral para los niveles por encima de los cuales desea mantener. Todos los demás valores se combinarán en "Otros".

Puedes cambiar la visualización del nivel "Otros" con other\_level =. A continuación, todos los hospitales excepto los dos más frecuentes se combinan en "Otros hospitales".

, warn ## Mostrar todos los niveles

Una de las ventajas del uso de factores es la estandarización del aspecto de las leyendas de los gráficos y de las tablas, independientemente de los valores que estén realmente presentes en unos datos.

Si está preparando muchas figuras (por ejemplo, para varias jurisdicciones), querrá que las leyendas y las tablas aparezcan de forma idéntica incluso con distintos niveles de compleción o composición de los datos.

### En los gráficos

En una figura ggplot(), basta con añadir el argumento drop = FALSE en la función scale\_xxxx() correspondiente. Se mostrarán todos los niveles de los factores, independientemente de si están presentes en los datos. Si sus niveles de columna de factores se muestran con fill =, entonces en scale\_fill\_discrete() incluya drop = FALSE, como se muestra a continuación. Si sus niveles se muestran con x = (al eje x) color = o tamaño =, deberá proporcionar esto a scale\_color\_discrete() o scale\_size\_discrete() según corresponda.

Este ejemplo es un gráfico de barras apiladas de la categoría de edad, por hospital. Añadiendo scale\_fill\_discrete(drop = FALSE) se garantiza que todos los grupos de edad aparezcan en la leyenda, aunque no estén presentes en los datos.

### En tablas

Tanto la **base** R table() como tabyl() de **janitor** mostrarán todos los niveles de los factores (incluso los no utilizados).

Si utiliza count() o summarise() de **dplyr** para hacer una tabla, añada el argumento .drop = FALSE para incluir los recuentos de todos los niveles del factor, incluso los no utilizados.

Lea más en la página de [tablas descriptivas](#descriptive-tables), o en la [documentación de scale\_discrete](https://ggplot2.tidyverse.org/reference/scale_discrete.html), o en la [documentación de count()](https://dplyr.tidyverse.org/reference/count.html). Puedes ver otro ejemplo en la página de rastreo de [contactos](#contact-tracing-1).

## Epiweeks

Por favor, consulta la extensa discusión sobre cómo crear semanas epidemiológicas en la página de [Agrupar datos](#grouping-data).   
Consulta también la página [Trabajar con fechas](#working-with-dates-1) para obtener consejos sobre cómo crear y dar formato a las semanas epidemiológicas.

### Epiweeks en un gráfico

Si tu objetivo es crear epiweeks para mostrarlos en un gráfico, puede hacerlo simplemente con floor\_date() de **lubridate**, como se explica en la página de [Agrupar datos](#grouping-data). Los valores devueltos serán del tipo Date con el formato YYYY-MM-DD. Si utiliza esta columna en un gráfico, las fechas se ordenarán correctamente de forma natural, y no tendrá que preocuparse de los niveles o de la conversión al tipo Factor. Vea el histograma ggplot() de las fechas de inicio más abajo.

En este enfoque, puede ajustar la visualización de las fechas en un eje con scale\_x\_date(). Consulta la página sobre [curvas epidémicas](#epidemic-curves) para obtener más información. Puedes especificar un formato de visualización "strptime" al argumento date\_labels = de scale\_x\_date(). Estos formatos utilizan marcadores de posición "%" y se tratan en la página [Trabajar con fechas](#working-with-dates-1). Utiliza "%Y" para representar un año de 4 dígitos, y "%W" o "%U" para representar el número de la semana (semana del lunes o del domingo respectivamente).

### Epiweeks en los datos

Sin embargo, si tu propósito al factorizar no es trazar, puede enfocar esto de dos maneras:

1. Para un control preciso de la visualización, convierta la columna de la epi-semana **lubrificada** (AAAA-MM-DD) al formato de visualización deseado (AAAA-WWw) dentro del propio dataframe, y luego conviértala en el tipo Factor.

En primer lugar, utiliza format() de para convertir la visualización de la fecha de YYYY-MM-DD a YYYY-Www (consulta la página [Trabajar con fechas](#working-with-dates-1)). En este proceso el tipo será convertida a carácter. A continuación, convierta de carácter a tipo Factor con factor().

**PELIGRO:** Si coloca las semanas por delante de los años ("Www-YYY") ("%W-%Y"), la ordenación por defecto del nivel alfanumérico será incorrecta (por ejemplo, 01-2015 estará antes que 35-2014). Podría ser necesario ajustar manualmente el orden, lo que sería un proceso largo y doloroso.

1. Para una visualización rápida por defecto, utiliza el paquete **aweek** y su función date2week(). Puedes establecer el week\_start = día, y si establece factor = TRUE entonces la columna de salida es un factor ordenado. Como ventaja, el factor incluye niveles para todas las semanas posibles en el lapso - incluso si no hay casos esa semana.

Consulta la página [Trabajar con fechas](#working-with-dates-1) para obtener más información sobre **aweek**. También ofrece la función inversa week2date().

## Recursos

Página de R for Data Science sobre [factores](https://r4ds.had.co.nz/factors.html)  
[viñeta del paquete aweek](https://cran.r-project.org/web/packages/aweek/vignettes/introduction.html)