Manejo de Datos con dplyr

Edgar Ramírez Galeano


tidyverse
www.rstudio.com
readr
ggplot2
forcats
tidyr
tidyr.tidyverse.org
TIBBLE
dplyr
dplyr.tidyverse.org
stringr
purrr

¿Qué es dplyr?

- **dplyr** es una librería de R para manipulación de datos.
- Facilita tareas comunes como filtrar, seleccionar, y transformar datos.
- Forma parte del paquete *tidyverse*.

Pipe

La sintaxis the pipe, de tuberías, permite expresar de forma clara una secuencia de múltiples operaciones. Es una sintaxis en cadena, de forma que el operador $\%>\%$ toma el output ('la salida') de una sentencia de código y la convierte en el input ('el argumento') de una nueva sentencia.

Principales Funciones de dplyr y ggplot

- `filter()`
- `select()`
- `mutate()`
- `arrange()`
- `summarize()`
- `group_by()`

filter()

- Filtra filas basadas en condiciones.

```
library(dplyr)
data %>% filter(condition)
```

mtcars % > %
filter(cyl==6 & hp>120)

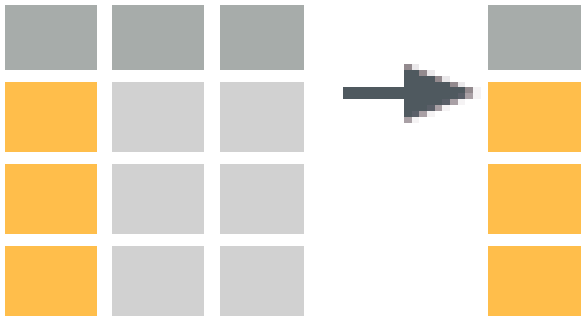
Logical and boolean operators to use with filter()

==	<	<=	is.na()	%in%	|	xor()
!=	>	>=	!is.na()	!	&	

select()

- Selecciona columnas específicas.

```
data %>% select(column1, column2)
```

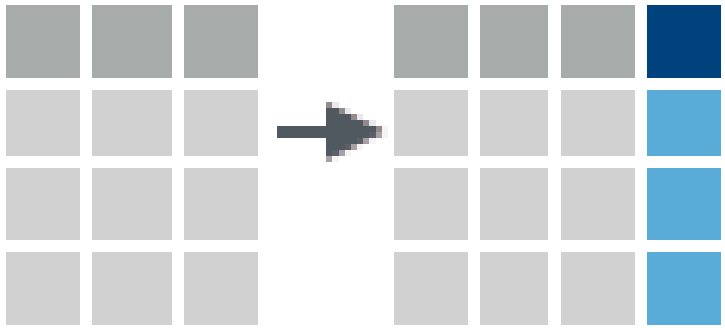


mtcars % > %
select(mpg,disp)

mutate()

- Crea nuevas columnas o modifica las existentes.

```
data %>% mutate(new_column = column1 * 2)
```

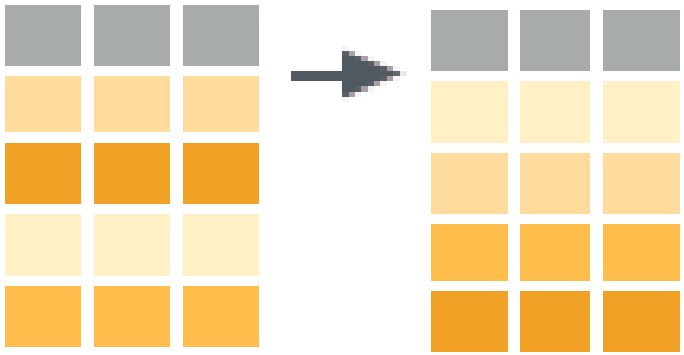


mtcars %>%
mutate(wtkg = (wt * 0.453592))

arrange()

- Ordena las filas según una o más columnas.

```
data %>% arrange(column1, desc(column2))
```



mtcars % > %
arrange(disp)

summarize() y group_by()

- Resume datos y realiza agregaciones.

```
data %>% group_by(category) %>%
summarize(mean_value = mean(value))
```

mtcars % > %
group_by(am) % > %
summarise(promedio=mean(mpg), max(mpg))

Ejemplo Completo

- Aplicar varias funciones en conjunto.

```
data %>%
filter(condition) %>%
select(column1, column2) %>%
mutate(new_column = column1 * 2) %>%
arrange(desc(new_column)) %>%
group_by(group_column) %>%
summarize(mean_value = mean(new_column))
```

Conclusión

- **dplyr** simplifica la manipulación de datos en R.
- Las funciones son intuitivas y eficientes.
- Integración fluida con el *tidyverse*.

Recursos Adicionales dplyr

- Documentación oficial de dplyr
- R for Data Science
- Sitio web del Tidyverse

¿Qué son los datos faltantes?

- Datos faltantes son valores ausentes en un conjunto de datos donde se esperaba una observación.
- Pueden ocurrir por errores en la recolección de datos, problemas técnicos, o porque los datos no existen.
- Representación en R: `NA` (Not Available).

Importancia de manejar datos faltantes

- Mantener la integridad de los análisis estadísticos.
- Asegurar la precisión de los modelos predictivos.
- Evitar sesgos en los resultados.
- Mejorar la calidad general de los datos.

¿Qué hacer con los datos faltantes?

- Retener los datos faltantes (en los casos en que no interfieren con el análisis).
- Eliminar la observación completa (toda la fila). Se analizan únicamente los casos sin datos faltantes..
- Imputar el dato faltante (reemplazarlo por otro valor).

Existen funciones que incluyen las funciones como argumento **“na.omit”** o **“na.rm = TRUE,”** que permiten ejecutar funciones numéricas sobre datos NA, algunos análisis devolverán un error ante la existencia de valores NA o vacíos ("").

Funciones para manejo de NA

- **is.na()**: Esta función presenta un valor lógico de TRUE si existe un valor ausente n una fila (funciona en vector, lista, matrices y data frame)..
- **na.omit()**: Elimina los registros que posean los valores "NA" de un conjunto de datos para su inclusión como argumento de otras funciones.

EJEMPLO

```
alumnos <- data.frame(nombre =c("Juan","Maria","Pedro","Luis"),
edad =c(NA,23,35,52))
is.na(alumnos)
mean(alumnos$edad , na.rm = T)
table(alumnos$edad , useNA = c("always"))
mean(na.omit(alumnos$edad))
drop_na(alumnos)
```

Métodos de Imputación

Imputación Simple

- Sustitución por la media o mediana
- Sustitución por un valor constante

replace_na() : Reemplazar NA con valores especificados

Técnicas de Imputación para Variables Categóricas

- **Sustitución por la Moda:** Reemplaza los valores faltantes con la categoría más frecuente (la moda) de la variable.
- **Sustitución por una Categoría Especial:** Reemplaza los valores faltantes con una nueva categoría, como "Desconocido" o "Faltante".
- **Imputación por Modelo:** Utiliza un modelo predictivo para predecir los valores faltantes basados en otras variables.

Reto dplyr

La base de datos **"Frijol.xlsx"**, utilizada para este estudio, contiene información sobre el uso de tecnología y servicios en el sector agrícola. Estos datos han sido obtenidos del Sistema de Información Agropecuaria y Pesquera (SIAP) de la SAGARPA, y corresponden a los años agrícolas 2017, 2018 y 2019.

¿Qué es ggplot2?

- ggplot2 es una librería de R para la creación de gráficos.
- Basada en la gramática de gráficos (The Grammar of Graphics).
- Facilita la creación de gráficos complejos de manera intuitiva.

Componentes Clave

- Datos (Data)
- Estética (Aesthetics, ‘aes‘): Especifica y las variables estéticas (x, y, color, etc.)
- Geometrías (Geometries, ‘geom‘): Diferentes tipos de gráficos: puntos, líneas, barras, etc.
- Facetas (Facets): Dividir el gráfico en subgráficos basados en una variable.
- Temas (Themes): Personalizar la apariencia del gráfico.

Objetos aestético

En ggplot2, aestético significa "algo que puedes ver". Algunos ejemplos son:

- Posición (por ejemplo, los ejes x e y)
- Color (color "externo")
- Fill (color de relleno)
- Shape (forma de puntos)
- Linetype (tipo de linea)
- Size (tamaño)

Cada tipo de objeto geométrico (geom) solo acepta un subconjunto de todos los aestéticos.

Objetos geométricos

Cada tipo de objeto geométrico (geom) solo acepta un subconjunto de todos los aestéticos.
Los objetos geométricos son las formas que puede tomar un gráfico. Algunos ejemplos son:

- Puntos (geom_point() para scatter plots, gráficos de puntos, etc. . .).
- Lineas (geom_line() para series temporales, lineas de tendencia, etc. . .).
- Cajas (geom_boxplot() para gráficos de cajas).

Un gráfico debe tener al menos un geom, pero no hay limite. Puedes añadir más geom usando el signo $+$.

geom_histogram

Un histograma es una representación gráfica de los datos agrupados en compartimentos o bins. Estos compartimentos incluyen individuos con
factores o agrupaciones de valores similares o cercanos numéricamente.

Histograma

```
ggplot(data=NutritionStudy)+
geom_histogram(aes(x=Age),bins=8, color="red", fill="white")
```

geom_freqpoly

Permite ver la distribución de una variable numérica, atenuando la curva. Se puede graficar mapeando una sola variable al menos.

Polígono de Frecuencias

```
ggplot(data=NutritionStudy)+
geom_freqpoly(aes(x=Age),bins=8,color="blue")
```

Histograma y Polígono de Frecuencias

```
ggplot(data=NutritionStudy)+
geom_histogram(aes(x=Age),bins=8, color="red", fill="white") +
geom_freqpoly(aes(x=Age),bins=8,color="blue")
```

geom_bar

Gráfico de barras es una representación de la frecuencia de ocurrencia de eventos para valores discretos; por ejemplo, sí o no, o valores categóricos.

Gráfico de barras

```
ggplot(data=NutritionStudy, aes(x=Smoke)) +
geom_bar(fill="white",color="blue")
```

Multiples grupos

```
ggplot(data=NutritionStudy, aes(Smoke,fill=Sex)) +
geom_bar(position = "dodge")
```

Para presentar los grupos uno al lado del otro se usa el parámetro position =“dodge”, lo que hará que cada grupo quede posicionado contiguamente.

geom_boxplot()

Hace gráficos boxplot, también conocidos como de bigote y caja.

BoxPlot

```
ggplot(data=NutritionStudy,aes(x=Age,y=Smoke)) +
geom_boxplot(color="green") +
coord_flip()
```

geom_point y geom_jitter

```
ggplot(data=NutritionStudy,aes(x=Age,y=Smoke)) +
geom_boxplot(color="green") +
geom_jitter(color="red",alpha=.1) +
coord_flip()
```

Con geom_point los datos iguales estarán completamente solapados. Por otro lado, en el gráfico de jitter los valores se han desplazado un poco.

geom_point

Gráficos de dispersión (también conocidos como scatter plots).

Scatter plots

```
ggplot(data=NutritionStudy,aes(x=Age,y=Cholesterol))+
geom_point(color="blue")
```

Añadir una tercera variable con color en los puntos

```
ggplot(data=NutritionStudy,aes(x=Age,y=Cholesterol,
colour=factor(PriorSmoke)))+
geom_point()
```

facet_wrap

Crear gráficos de paneles, también llamados gráficos de Trellis o facetas.

```
ggplot(data=NutritionStudy,aes(x=Age,y=Cholesterol,
colour=Sex))+
geom_point() +
facet_wrap(~PriorSmoke)
```

```
ggplot(data=NutritionStudy,aes(x=Age,y=Cholesterol,
colour=VitaminUse))+
geom_point() +
facet_grid(Sex~PriorSmoke)
```

Etiquetas

```
ggplot(data=NutritionStudy, aes( x=Age, y=Cholesterol,
colour=factor(PriorSmoke)))+
geom_point() +
labs(title="Gráfico Nutrición",
subtitle = "Edad vs Colesterol",
x="Edad",
y="Colesterol",
colour="Est Tabaquismo")
```

Temas

En ggplot, existen numerosos temas por defecto para elegir.

- theme_gray(): Fondo gris y líneas de cuadrícula blancas.
- theme_bw(): El tema clásico de ggplot2 con fondo oscuro. Puede funcionar mejor para presentaciones mostradas con un proyector.
- them_linedraw(): Un tema con sólo líneas negras de varios anchos sobre fondos blancos, que recuerda a un dibujo de líneas.
- theme_dark(). El primo oscuro de theme_light(), con tamaños de línea similares pero con un fondo oscuro. Es útil para hacer que las líneas finas de color resalten.
- theme_minimal(). Un tema minimalista sin anotaciones de fondo.
- theme_classic(). Un tema de aspecto clásico, con líneas en los ejes X e Y y sin líneas de cuadrícula.

Reto ggplot

Realizar los gráficos utilizando la librería ggplot de R utilizando los datos HEART contienen una variable binaria (HD) de 303 pacientes que se presentaron con dolor en el pecho.

Un valor de resultado de “Yes” indica la presencia de enfermedad del corazón basado en una prueba angiografica, mientras que “No” significa que no hay enfermedades del corazón.