¡Vamos a seguir avanzando!

Ya has visto que los paquetes esenciales de R son:

Manipular data frames:
- dplyr (tidyverse) - https://dplyr.tidyverse.org/
- tidyr (tidyverse) - https://tidyr.tidyverse.org/

Crear gráficos:
- ggplot2 (tidyverse) - https://ggplot2.tidyverse.org/ - https://r-graph-gallery.com/
- plotly - https://plotly.com/r/
- ggpubr (plots para publicaciones) - https://rpkgs.datanovia.com/ggpubr/
- ggstatsplot (plots para publicaciones) - https://indrajeetpatil.github.io/ggstatsplot/

Paquetes estadísticos:
- Hmisc (many functions useful for data analysis) - https://hbiostat.org/R/Hmisc/examples.html
- corrplot (correlograma) - http://www.sthda.com/english/wiki/visualize-correlation-matrix-using-correlogram
- stats (base de R no hace falta instalar) - https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html

Paquetes machine learning:
- caret (machine learning supervisado) - https://topepo.github.io/caret/

Hay más pero los esenciales podrían ser estos.

En esta hoja de trabajo quiero que practiques principalmente cómo manipular data frames y cómo crear gráficos.

Pero antes te obligaré a crear un repositorio o lista de enlaces clave de recursos de consulta de R.

¡ A por ello!

1. **Crea un documento con la lista de url de consulta de R**

- Crea un documento con la lista de enlaces de consulta de temas de R. Aquí ya tienes unos cuántos enlaces para empezar esa lista. Guarda esta lista en tu carpeta R que has creado

2. **Ejercicios de manipular data frames:**

- Lee el data frame mtcars usando el comando data <- mtcars
- Haz un summary de los datos
- Crea un data frame con: nombres_variables = data.frame(Nombre = colnames(data),Num_Columna = seq(ncol(data)))
- Selecciona en un data frame llamado df_sel las filas de la 5 a la 10 y las 5 últimas variables
- Selecciona solo los coches con un mpg más grande que 25

- Selecciona solo los coches con un mpg más grande que 25 y am = 1
- Ordena los coches de mayor a menor en qsec y que sean de am= 0 y gear = 4. ¿Qué coche tiene más qsec?
- Calcula el promedio de wt (peso) por cyl y am

Apunta aquí:

¿Qué coche tiene más qsec del grupo de am = 0 y gear = 4?

El coche que más qsec tiene del grupo am = 0 y gear = 4 es MERC 230 con 22.9

La tabla de promedios de wt (peso) por cyl y am

	cyl	am	Mean_wt
1	4	0	2.935000
2	4	1	2.042250
3	6	0	3.388750
4	6	1	2.755000
5	8	0	4.104083
6	8	1	3.370000

DUDA:
Realizando el ejercicio del cálculo de los promedios de «wt» en función de «cyl» y «am».
En tu vídeo cuando explicabas esta parte, comentabas que para usar la función «group_by» la variable de agrupación debe ser un factor. Sin embargo en este caso ni «cyl» ni «am» lo son. ¿Tendría que transformarlas antes? Cuando lo he corrido, no me ha saltado ningún warning (que sin embargo más adelante para hacer los gráficos, sí que me ha saltado y he tenido que transformar «am» en un factor); me resulta un poco confuso que unas veces sí y otras no, o yo no estoy viendo la diferencia, ¿me lo aclaras?
RESPUESTA:
Entiendo tu confusión. En R, la función group_by() de dplyr puede trabajar con variables tanto factor como no factor (como números o cadenas de texto). La confusión puede surgir porque el tratamiento de las variables en R puede cambiar dependiendo del contexto y de la función específica que estés utilizando.

Veamos en detalle:

Uso de group_by con Factores y No Factores: La función group_by() agrupa los datos basándose en los valores únicos de las variables especificadas. No es necesario que estas variables sean factores; pueden ser de cualquier tipo, como numéricas o caracteres. En tu caso, aunque cyl y am no sean factores, group_by() funcionará correctamente agrupando los datos según los valores únicos de estas variables.

¿Por qué a veces es necesario convertir a factores? En algunas funciones y tipos de análisis, especialmente en gráficos, la conversión a factores puede ser útil o necesaria. Por ejemplo, en la creación de gráficos, convertir una variable en factor permite un mejor control sobre el orden y la representación de los grupos en el eje. Además, en análisis estadísticos, convertir variables numéricas en factores puede cambiar la forma en que se tratan en el modelo (por ejemplo, como categorías en lugar de valores continuos).

Ejemplo de group_by en tu Caso: En tu caso específico, estás calculando promedios de wt agrupados por cyl y am. Aquí no necesitas convertir estas variables en factores para que group_by() funcione. El código podría verse así:

R
Copy code
library(dplyr)

 # Asumiendo que estás trabajando con el dataset 'mtcars'

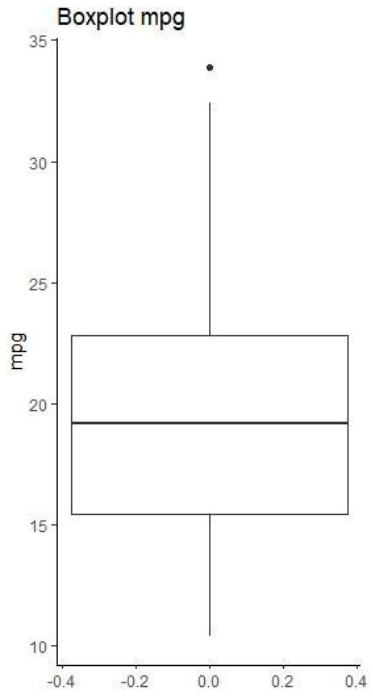
mtcars %>%
group_by(cyl, am) %>%
summarize(mean_wt = mean(wt))
Este código agrupará los datos por los valores únicos de cyl y am (sin importar si son factores o no) y calculará la media de wt para cada grupo.

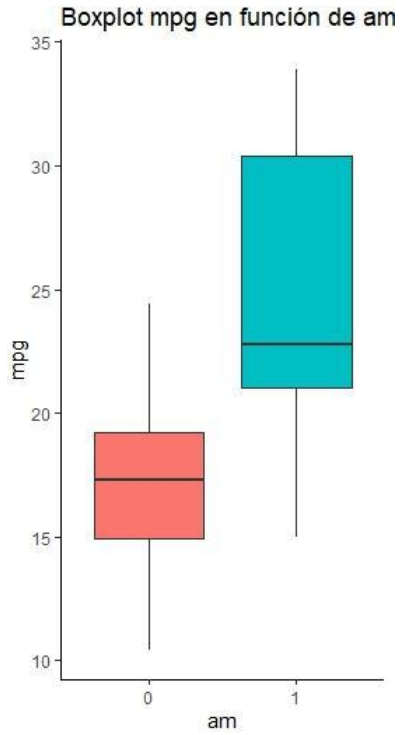
Recuerda que R es un lenguaje flexible y las necesidades de conversión a factores pueden variar según el contexto y la función que estés utilizando. En la práctica, si tu código funciona correctamente sin warnings y los resultados son los esperados, es probable que estés utilizando las funciones de manera adecuada para tus datos.

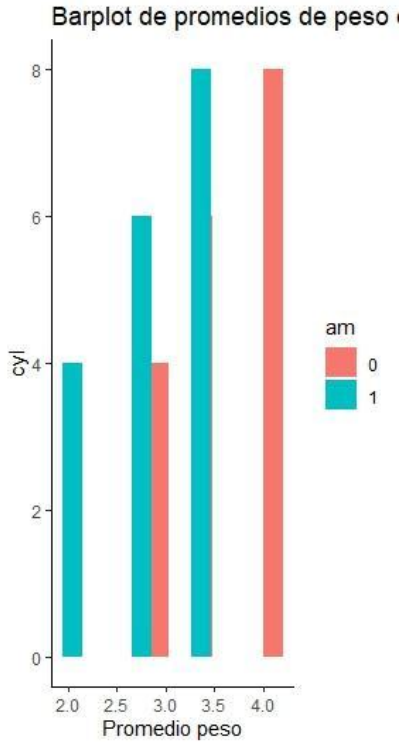
3. Ejercicios de crear gráficos:

- Lee el data frame mtcars usando el comando data <- mtcars
- Crea un boxplot de mpg usa esta referencia: http://www.sthda.com/english/wiki/ggplot2-box-plot-quick-start-guide-r-software-and-data-visualization
- Crea un boxplot de mpg por grupos de am, usa esta referencia: http://www.sthda.com/english/wiki/ggplot2-box-plot-quick-start-guide-r-software-and-data-visualization
- Calcula el promedio de wt (peso) por cyl y am
- Crea un barplot de los promedios anteriores, usa esta referencia: http://www.sthda.com/english/wiki/ggplot2-barplots-quick-start-guide-r-software-and-data-visualization

Copia aquí los gráficos que has creado:




Boxplot mpg en función de am
mpg
35
30
25
20
15
10
0
1
am


Barplot de promedios de peso
cyl
8
6
4
2
0
2.0
2.5
3.0
3.5
4.0
Promedio peso
am
0
1

DUDA**: **Este último plot me ha costado un poco más, no se si es exactamente esto lo que hay que representar. De hecho en la barra de 3.5 hay un overlapping de los dos grupos. R me daba el warning pero no sabía cómo quitarlo. Te adjunto el código para que lo revises me digas cómo subsanar el error:

```
library(dplyr)
data_sel2 = data %>%
  group_by(cyl,am) %>%
  summarize(Mean_wt = mean(wt))

library(ggplot2)
p3 <- ggplot(data_sel2, aes(x=Mean_wt, y=cyl, fill=am)) +
  geom_bar(stat="identity", width=0.2) +
  labs(title="Barplot de promedios de peso en función de cyl y am", x="Promedio peso") +
  theme_classic()
p3
```

RESPUESTA:

Tu código para generar el gráfico de barras parece estar bien estructurado, pero el problema que mencionas de «overlapping» (sobrelapamiento) de las barras es común cuando se utilizan gráficos de barras para mostrar múltiples grupos en las mismas categorías.

El problema aquí es que estás usando barras colocadas en las mismas posiciones del eje Y para diferentes grupos de am. Una forma común de abordar esto es usar barras agrupadas o apiladas. Sin embargo, dado que estás mapeando y=cyl y fill=am, la mejor opción podría ser un gráfico de barras agrupadas, donde las barras para diferentes valores de am están una al lado de la otra en lugar de superponerse.

Aquí te dejo una sugerencia para modificar tu código:

Usar Posición Ajustada: Puedes ajustar la posición de las barras para que se coloquen una al lado de la otra usando position = position_dodge() en geom_bar().

Ajustar la Anchura de las Barras: Puedes necesitar jugar un poco con el parámetro width en geom_bar() y position_dodge() para obtener una presentación visual que te satisfaga.

El código modificado sería algo así:

```
R
Copy code
library(dplyr)
library(ggplot2)

data_sel2 = data %>%
group_by(cyl, am) %>%
summarize(Mean_wt = mean(wt))

p3 <- ggplot(data_sel2, aes(x=as.factor(cyl), y=Mean_wt, fill=as.factor(am))) + geom_bar(stat="identity", position=position_dodge(width=0.8), width=0.7) + labs(title="Barplot de promedios de peso en función de cyl y am", x="Cilindrada", y="Promedio peso") + theme_classic()
p3
```

En este código, position_dodge(width=0.8) ayuda a separar las barras para diferentes valores de am dentro de cada categoría de cyl. El parámetro width=0.7 controla la anchura de las barras individuales. Puedes ajustar estos valores según sea necesario para mejorar la visualización. Además, he convertido cyl y am en factores en el mapeo aes(), ya que esto suele ser más adecuado para variables categóricas en un gráfico de barras.

¡Genial!

Ya has generado manipulado data frames y creado tus primeros gráficos usando referencias en internet.

¡Vas avanzando muy bien!

Celébralo como más te guste :)
Tómate una cerveza, un zumo, un té o lo que te guste. Son tus primeros pasos con R.

1 abrazo y hasta pronto.
Brindo por tus éxitos :)

Jordi Ollé