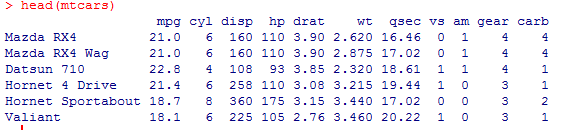
# Ejercicio 1

## Data frame mtcars

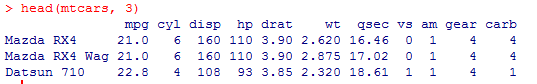
Con el data frame mtcars (viene cargado en R).

1. Previsualizar el contenido con la función head().

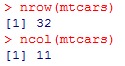
head(mtcars) # nos muestra los primeros registros del data frame. En este caso, al no decir cuantos, nos muestra 6



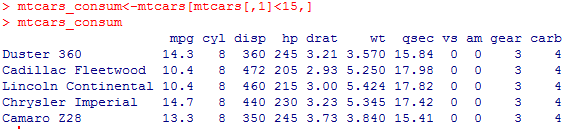
head(mtcars, 3) # nos muestra los tres primeros registros del data frame



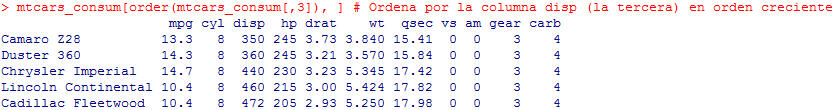
2. Mirar el número de filas y columnas con nrow() y ncol().



3. Crear un nuevo data frame con los modelos de coche que consumen menos de 15 millas/galón.



4. Ordenar el data frame anterior por disp.

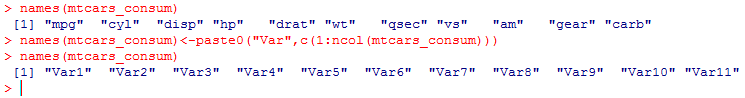


5. Calcular la media de las marchas (gear) de los modelos del data frame anterior.



6. Cambiar los nombres de las variables del data frame a var1, var2, ..., var11.

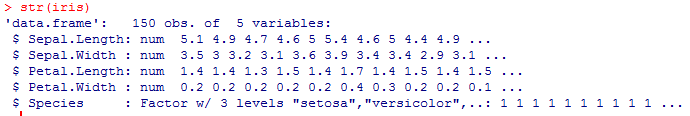
Pista: Mirar la documentación de la función paste y usarla para generar el vector (\var1", \var2", ..., \varN") donde N es el número de variables del data frame.



## Data frame Iris

1. ¿Cómo está estructurado el data frame? (utilizar las funciones str() y dim()).

str(iris) # nos da informacion del data frame: número de registros, número de variables, descripción de cada variable (nombre, tipo)



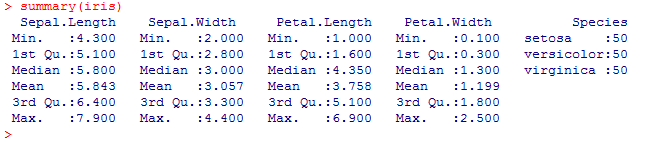
dim(iris) # nos devuelve la dimension del dataframe, número de filas x número de columnas



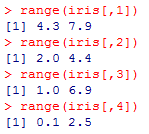
2. ¿De qué tipo es cada una de las variables del data frame?

Según el resultado devuelto por la instrucción str(iris) Las 4 primeras variables son de tipo numéric y la última (Species) es categórica, con tres categorías (setosa, versicolor y virginica

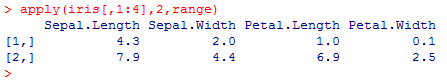
3. Utilizar la función summary() para obtener un resumen de los estadísticos de las variables.



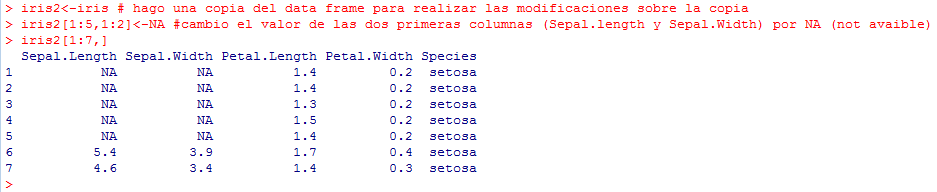
4. Comprobar con las funciones mean(), range(), que se obtienen los mismos valores.



Otra forma de hacerlo sería con la función apply():

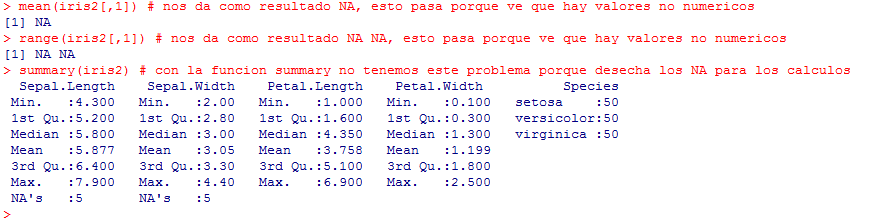


5. Cambia los valores de las variables Sepal.Length Sepal.Width de las 5 primeras observaciones por NA



He creado el data frame iris2 como copia de iris para realizar los cambios sin modificar el original.

6. ¿Qué pasa si usamos ahora las funciones mean(), range() con las variables Sepal.Length y Sepal.Width? ¿Tiene el mismo problema la función summary()?



Al utilizar las funciones mean() y range() se tienen en cuenta los valores NA, por lo que no se pude hacer el cálculo por no ser todos los elementos numéricos.

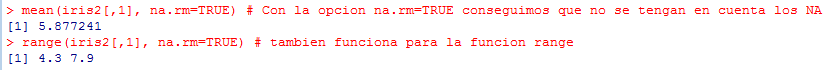
La función range() no tiene en cuenta estos valores para realizar el cálculo.

7. Ver la documentación de mean(), range(), etc. ¿Qué parámetro habrá que cambiar para arreglar el problema anterior?

Para ver la documentación tenemos dos opciones:



Después de ver la documentación, vemos que con la opción na.rm=TRUE conseguimos que no se tengan en cuenta los NA para los cálculos:



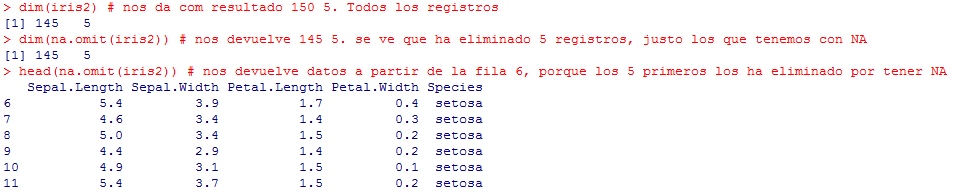
Vemos que los resultados son los mismos que con la función summary().

8. Visto lo anterior, ¿por qué es importante codificar los missing values como NA y no como 0, por ejemplo?

Al codificar como NA se pueden eliminar del cálculo de estadísticos para que no den resultados erroneos (o error). Si los codificamos como 0 o algún otro valor, entrarán en los cálculos dando lugar a resultados incorrectos.

9. Eliminar los valores NA usando na.omit().

na.omit(iris2) # nos devuelve el data frame iris2 sin los casos con NA



Para tener guardado el array sin los NA se lo asignamos a sí mismo:



10. Calcular la media de la variable Petal.Length para cada uno de las distintas especies (Species). Pista: usar la función tapply().

