
ATTACH & DETACH FUNCTIONS

- find what attach() and detach() commands do???

The attach function allows to access variables of a data.frame without calling the data.frame.

The detach function can be used to remove the attachment of a data.frame, which was previously attached with the attach function.

HEAD & TAIL FUNCTIONS

- find what head() and tail() commands do
Head(): Function which returns the first n rows of the dataset.
Tail(): Function which returns the last n rows of the dataset.
- Use head() and tail() commands to display sample observations of mtcarsdataset

```
head(mtcars)
```

```
tail(mtcars)
```

- Use head() command to Print first 10 observations

```
df<-datasets::airquality  
head(df,n=10)
```
- Use tail() commands to Print last 15 observations

```
df<-datasets::airquality  
tail(df,n=15)
```

SORTING

- Sort the observations of the dataset “mtcars” in increasing order based on the values in the column "mpg"

```
newdata <- mtcars[order(mpg),]
```

- Sort the observations of the dataset “mtcars” in decreasing order based on the values in the column "cyl"

```
newdata <- mtcars[order(-cyl),]
```

- Sort the observations of the dataset “mtcars” in increasing order based on the values in the columns both "mpg" and "cyl"

```
newdata <- mtcars[order(mpg, cyl),]
```

- Sort the observations of the dataset “mtcars” in decreasing order based on the values in the columns both "mpg" and "cyl"

```
newdata <- mtcars[order(-mpg, -cyl),]
```

- Sort the observations of the dataset “mtcars” by column “mpg” in increasing order and column “cyl” in decreasing order

```
newdata <- mtcars[order(mpg, -cyl),]
```

EXERCISES

- Create a vector as `x <- c(9:20, 1:5, 3:7, 0:8)`
`> x<-c(9:20,1:5,3:7,0:8)`
- Use `duplicated()` function to print the logical vector indicating the duplicate values present in `x`
`> duplicated(x)`
- Observe the output of `duplicated(x, fromLast = TRUE)`
- What is the difference between `duplicated(x)` and `duplicated(x,fromLast=TRUE)`
`duplicated(x,fromLast=TRUE)` is equivalent to but faster than `duplicated(x)`
- Extract duplicate elements from `x`
`> x[duplicated(x)]`
- Extract unique elements from `x`
`> x[!duplicated(x)]`
- Print duplicate elements from `x` in different order (**Hint:** Use `duplicated(x, fromLast = TRUE)`)
`> x[duplicated(x, fromLast = TRUE)]`
- Extract unique elements from `x` in different order (**Hint:** Use `duplicated(x, fromLast = TRUE)`)
`> x[!duplicated(x, fromLast = TRUE)]`
- Print the indices of duplicate elements
`> which(duplicated(x))`
- Print the indices of unique elements
`> which(!duplicated(x))`
- How many unique elements are in `x`
`> sum(!duplicated(x))`
- How many duplicate elements are in `x`
`> sum(duplicated(x))`
- Create a dataframe `df` :
`a <- c(rep("A", 3), rep("B", 3), rep("C",2))`
`b <- c(1,1,2,4,1,1,2,2)`
`df <- data.frame(a,b)`
- Use `duplicated()` function to print the logical vector indicating the duplicate values present in dataframe “`df`”
`> duplicated(df)`

- Extract duplicate elements from dataframe “df”
> df[duplicated(df),]
- Extract unique elements from dataframe “df”
> df[!duplicated(df),]
- Print the indices of duplicate elements
> which(duplicated(df))
- Print the indices of unique elements
> which(!duplicated(df))
- How many unique elements are in dataframe "df"
> sum(!duplicated(df))
- How many duplicate elements are in dataframe "df"
> sum(duplicated(df))

EXERCISES

- Print the dataset *iris*
> iris
- Print the structure of the dataset *iris*
> str(iris)
- Print the summary of all the variables of the dataset *iris* (**Hint:** Use function *summary()*)
> summary(iris)
- How many of the variables (columns) are in the dataset *iris*
> ncol(iris)
- How many observations (rows) are in the dataset *iris*
> nrow(iris)
- Use *duplicated()* function to print the logical vector indicating the duplicate values present in the dataset *iris*
> duplicated(iris)
- Extract duplicate elements from the dataset *iris*
> iris[duplicated(iris),]
- Extract unique elements from the dataset *iris*
> iris[!duplicated(iris),]

- Print the indices of duplicate elements in the dataset *iris*
`> which(duplicated(iris))`
- Print the indices of unique elements in the dataset *iris*
`> which(!duplicated(iris))`
- How many unique elements are in the dataset *iris*
`> sum(!duplicated(iris))`
- How many duplicate elements are in the dataset *iris*
`> sum(duplicated(iris))`

----- **EXERCISES** -----

- Practice above examples that generate NA values
- Create NA values by some illegal operations
`> as.numeric (c("1", "2", "three", "4"))`
- Practice exercises in lecture slide
- What happens when we try to sort the data with NA values
 By default, sort removes any NA values and can therefore change the length of a vector.
- How to find the length of a vector with NA values
`length(x)`
`length(x) <- n`