Data Structures in R: Data Frames part 2

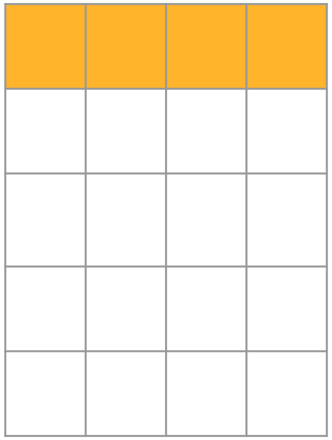
Stat 133 with Gaston Sanchez

Basic manipulation of Data Frames

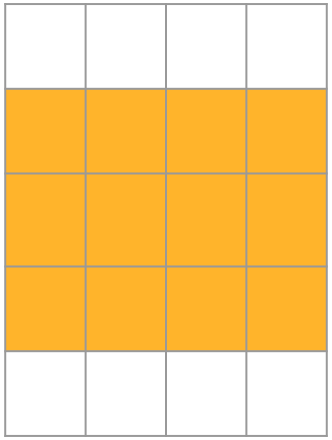
Working with data frames

There are many ways in which the elements of a data.frame can be accessed (i.e. retrieved, selected)

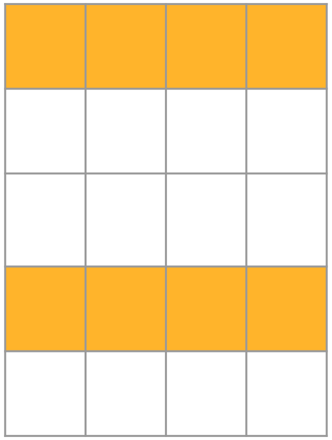
Accessing Rows



one single
row

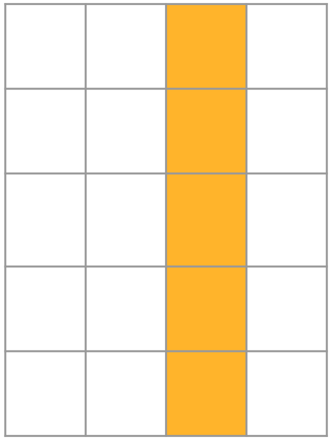


consecutive
rows

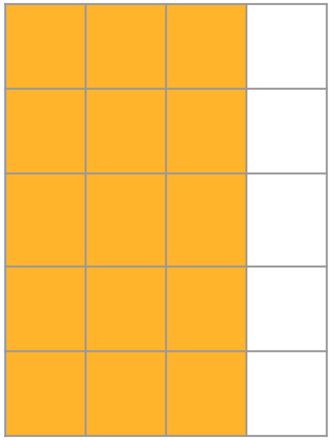


separate
rows

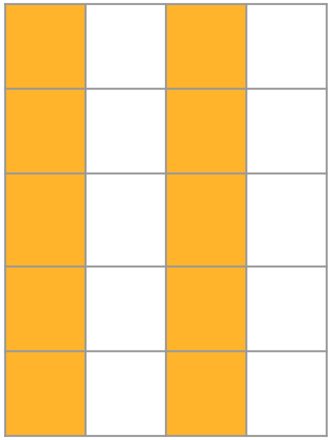
Accessing Columns



one single column

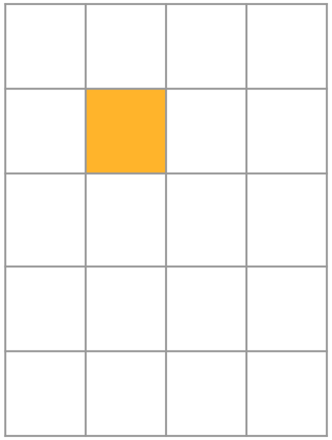


consecutive columns

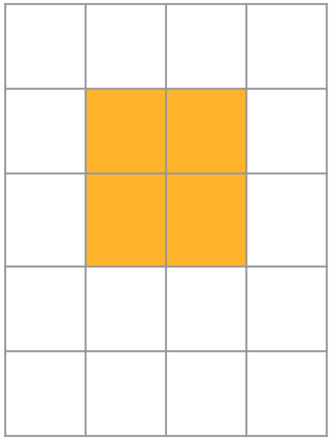


separate columns

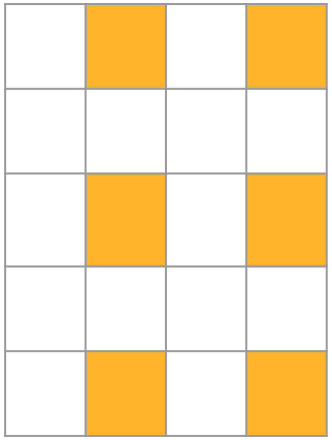
Accessing Cells



one single
cell



consecutive
cells



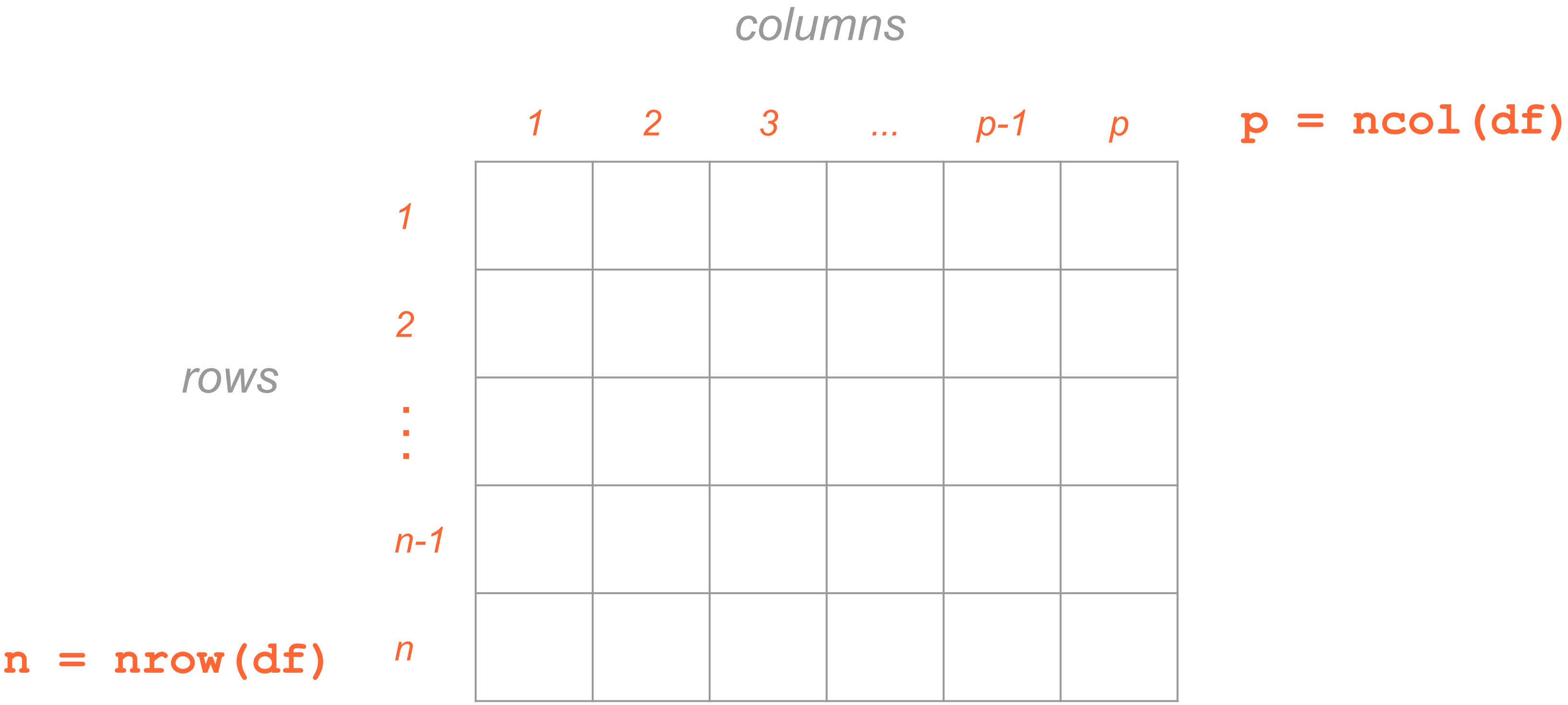
separate
cells

Data frame `airquality` *(first 10 rows)*

```
   Ozone Solar.R Wind Temp Month Day
1     41     190  7.4   67     5   1
2     36     118  8.0   72     5   2
3     12     149 12.6   74     5   3
4     18     313 11.5   62     5   4
5     NA      NA 14.3   56     5   5
6     28      NA 14.9   66     5   6
7     23     299  8.6   65     5   7
8     19      99 13.8   59     5   8
9      8      19 20.1   61     5   9
10    NA     194  8.6   69     5  10
```

Retrieving elements via Index Values

Numeric Indices in a data frame



columns

1 *2* *3* *...* *p-1* *p*

`p = ncol(df)`

rows

1

2

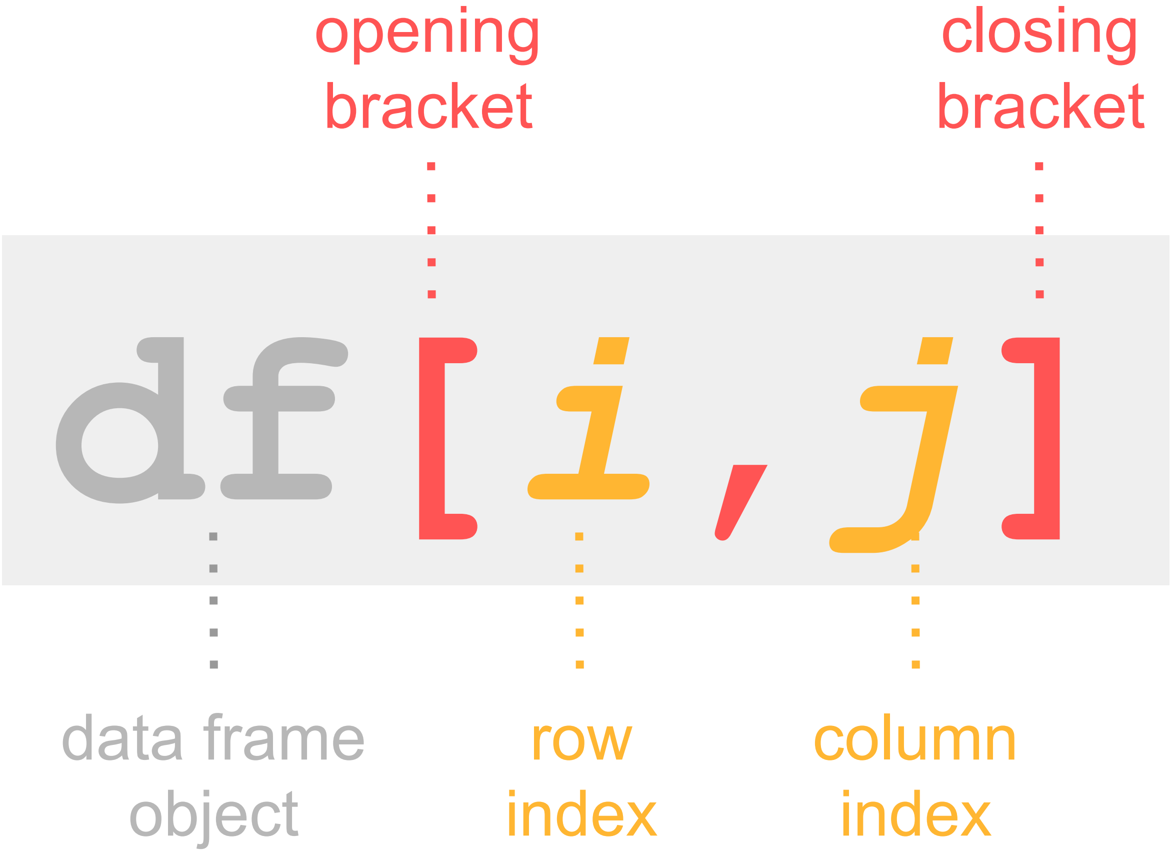
⋮

n-1

n

`n = nrow(df)`

Bracket Notation



opening bracket

closing bracket

```
df[i,j]
```

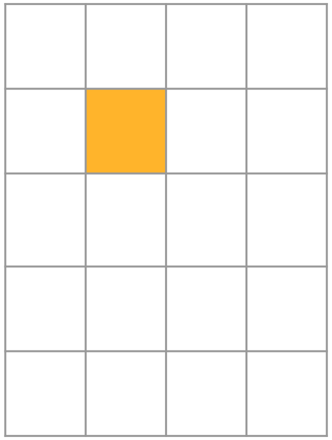
data frame object

row index

column index

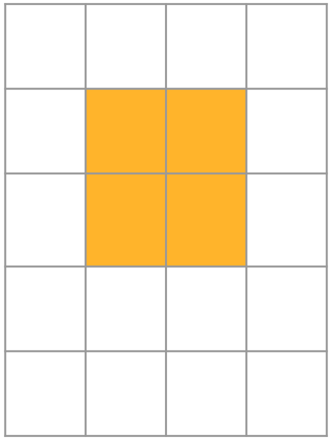
Retrieving Cells

`df[2,2]`



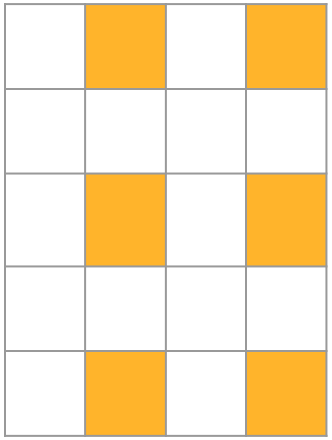
one single
cell

`df[2:3,2:3]`



consecutive
cells

```
df[c(1,3,5),
   c(2,4)]
```



separated
cells

Retrieving Cells

```
# first cell 1,1
airquality[1,1]

# cell 9,6
airquality[9,6]

# last cell
airquality[153,6]
```

Retrieving Cells

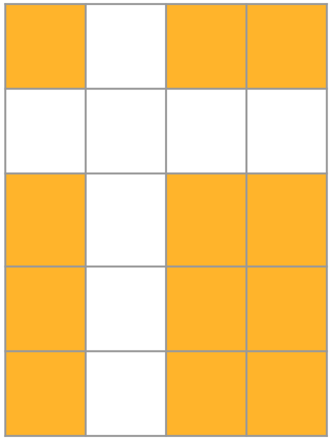
```
# various adjacent cells
airquality[1:5,4:6]

# various adjacent cells
#(permuted order)
airquality[5:1,6:4]

# non-adjacent cells
airquality[c(1,50,100),c(3,5)]
```

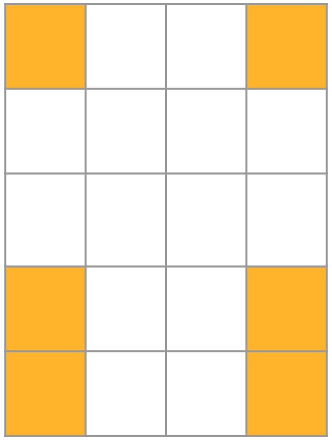
Retrieving Cells (excluding indices)

```
df[-2,-2]
```



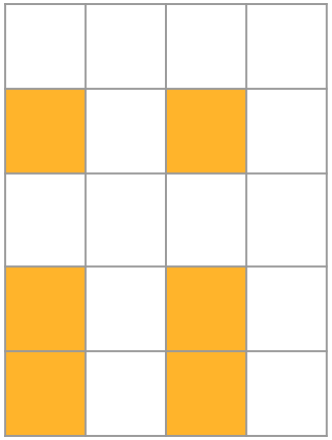
one single
cell

```
df[-(2:3),
   -(2:3)]
```



consecutive
cells

```
df[-c(1,3),
   -c(2,4)]
```



separated
cells

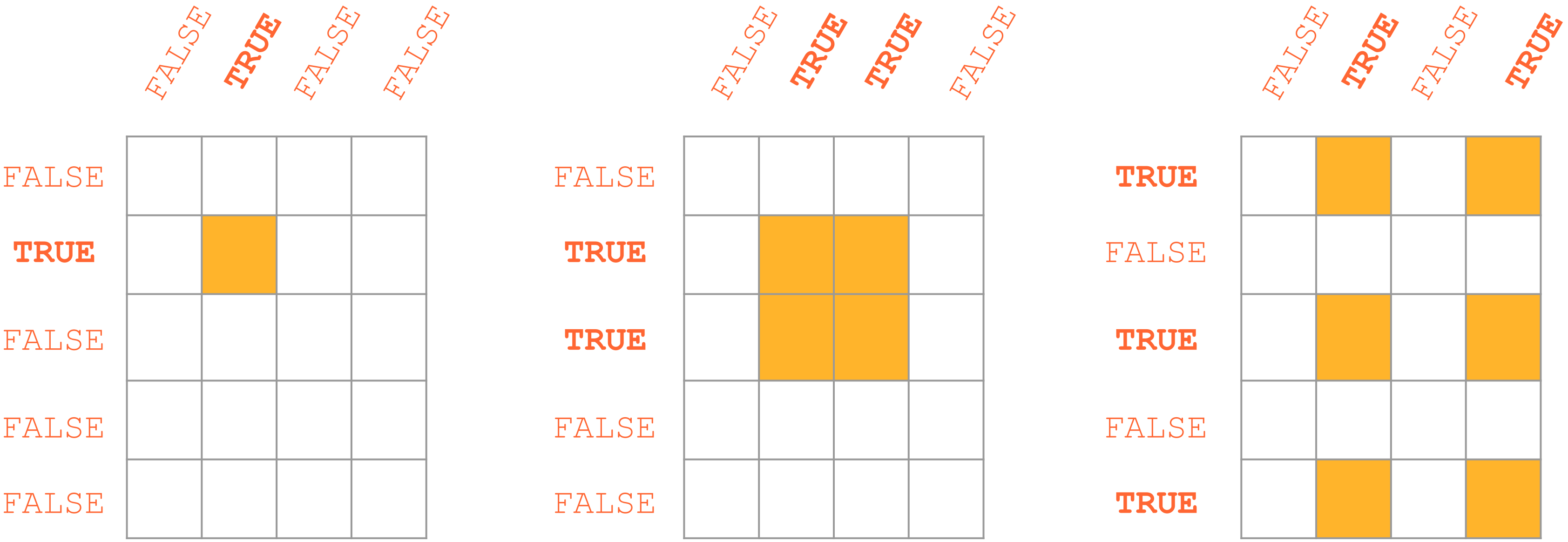
Retrieving Cells (excluding indices)

```
# various adjacent cells
airquality[-(1:5),-(4:6)]

# non-adjacent cells
airquality[-c(1,50,100),-c(3,5)]
```

Accessing Cells via Logical Subscripts

```
df[ilog,jlog]
```



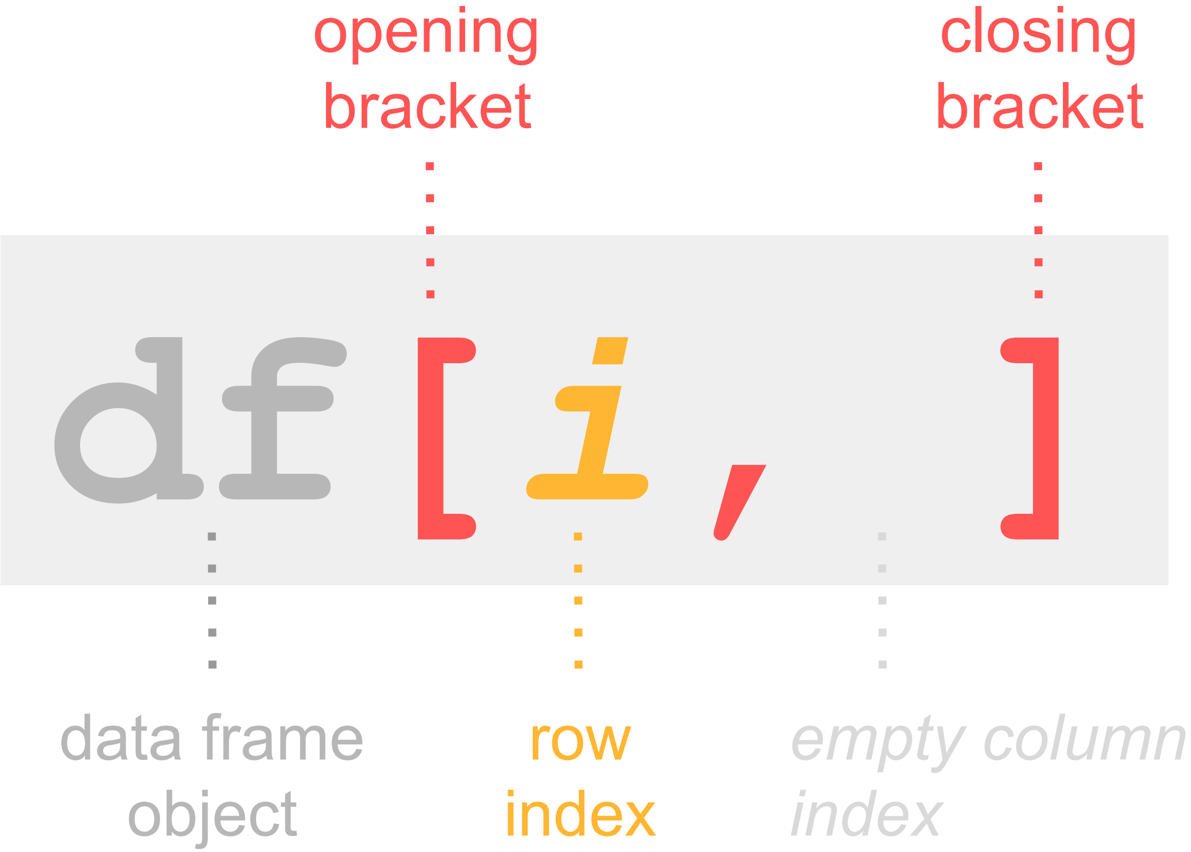
	FALSE	**TRUE**	FALSE	FALSE
FALSE				
TRUE		■		
FALSE				
FALSE				
FALSE				

	FALSE	**TRUE**	**TRUE**	FALSE
FALSE				
TRUE		■	■	
TRUE		■	■	
FALSE				
FALSE				

	FALSE	**TRUE**	FALSE	**TRUE**
TRUE		■		■
FALSE				
TRUE		■		■
FALSE				
TRUE		■		■

Gaston Sanchez

Bracket Notation: retrieving rows



opening bracket

closing bracket

```
df[i, ]
```

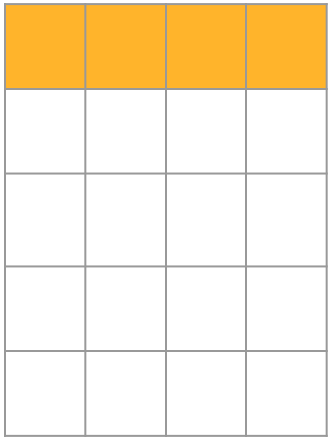
data frame object

row index

empty column index

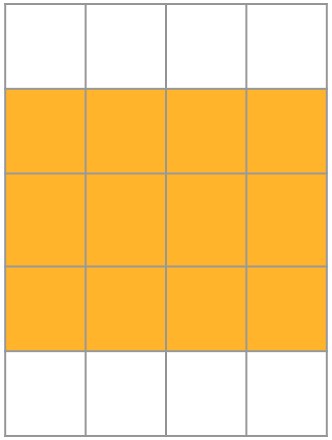
Retrieving Rows

`df[1, ]`



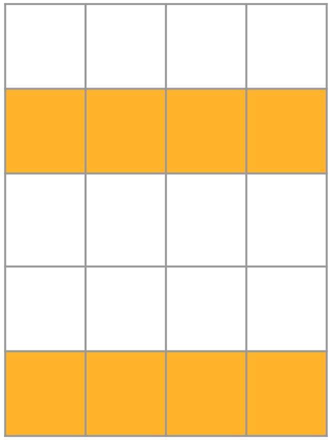
one single
row

`df[2:4, ]`



consecutive
rows

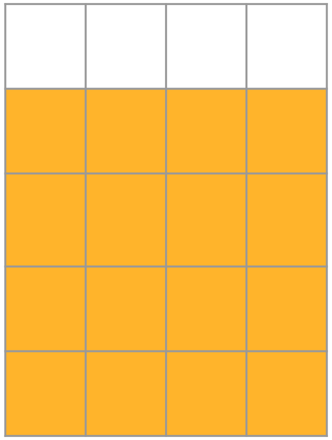
`df[c(2,5), ]`



separate
rows

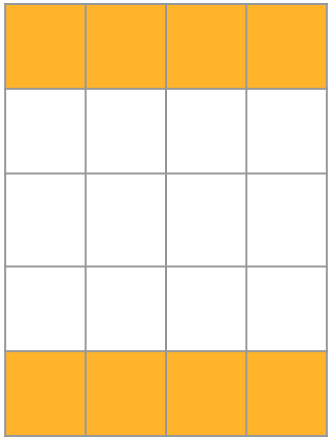
Retrieving Rows (excluding indices)

`df[-1, ]`



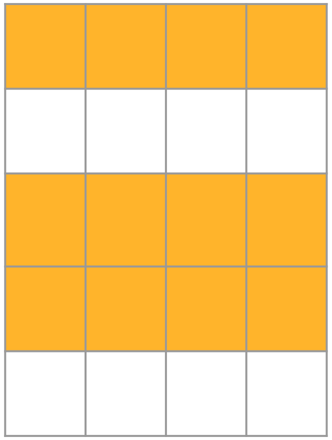
one single
row

`df[-(2:4), ]`



consecutive
rows

`df[-c(2,5), ]`



separate
rows

Retrieving Rows

```
# first row
airquality[1, ]

# rows from 3 to 7
airquality[3:7, ]

# rows 1, 3, 5, 7
airquality[c(1,3,5,7), ]
```

Retrieving Rows (excluding indices)

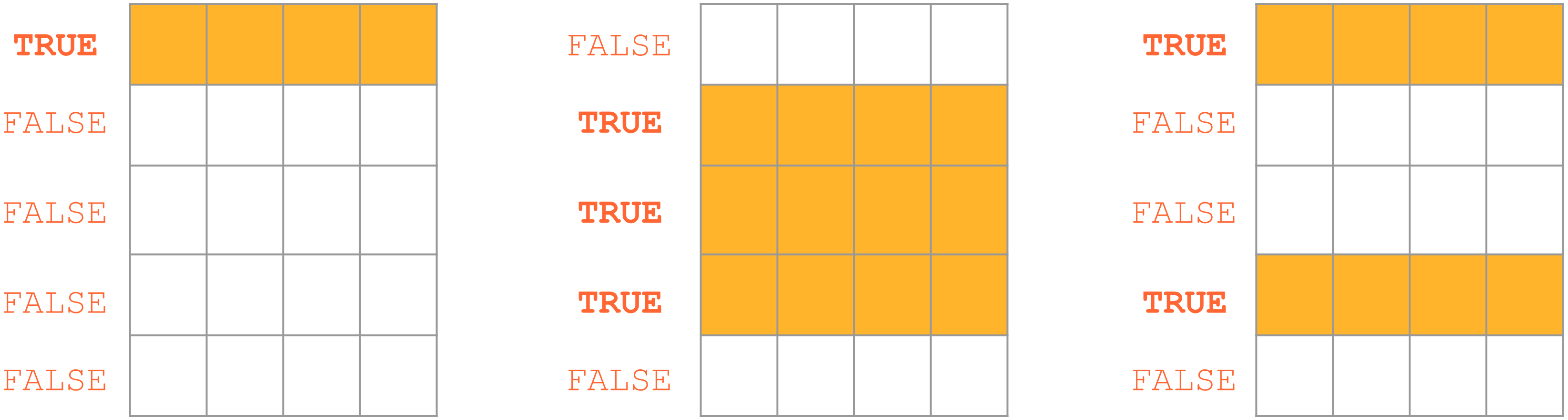
```
# all rows except first one
airquality[-1, ]

# rows except from 3 to 7
airquality[-(3:7), ]

# all rows but 1, 3, 5, 7
airquality[-c(1,3,5,7), ]
```

Accessing Rows via Logical Subscripts

```
df[logical, ]
```



TRUE
FALSE
FALSE
FALSE
FALSE

FALSE
TRUE
TRUE
TRUE
FALSE

TRUE
FALSE
FALSE
TRUE
FALSE

Retrieving Rows (logical indexing)

```
# records with Month 5
airquality[airquality$Month==5, ]

# records of 1st day of month
airquality[airquality$Day==1, ]
```

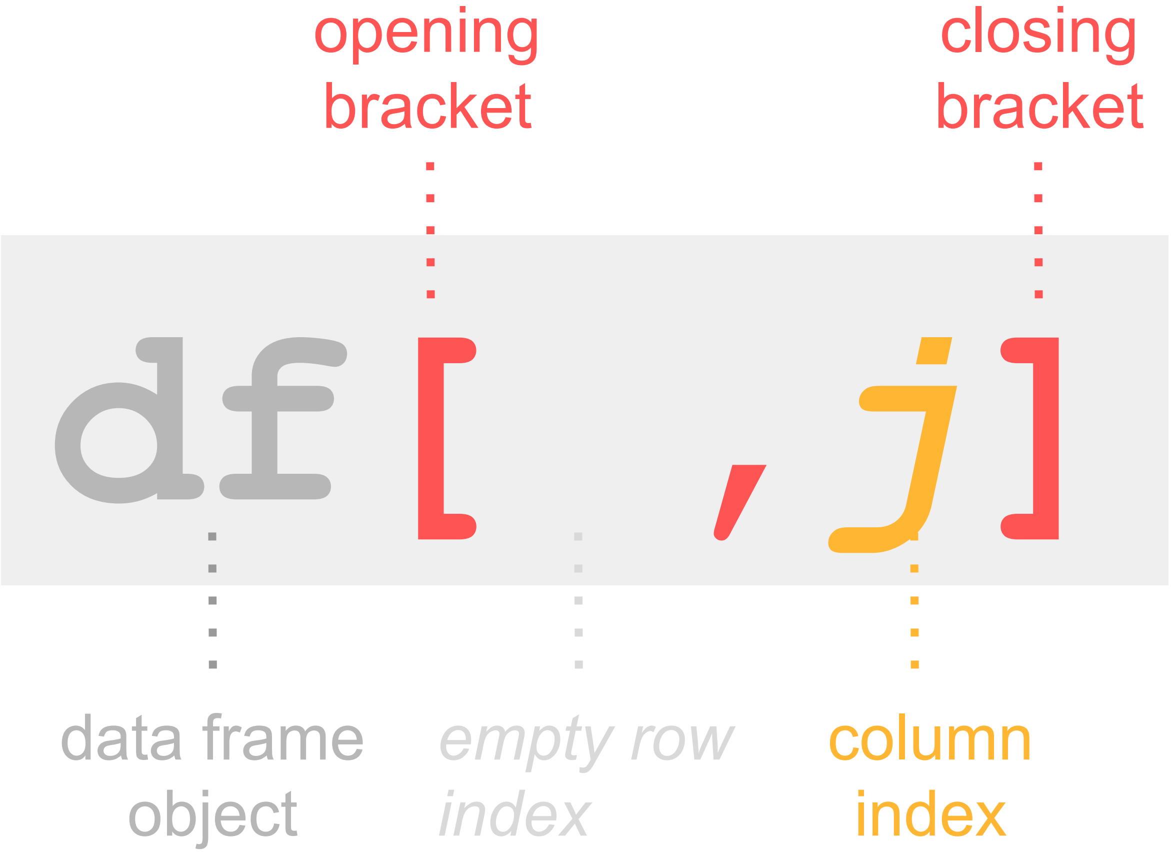
Retrieving Rows (logical indexing)

```
# vector matching odd numbers
odds = rep(c(TRUE, FALSE),
  length = nrow(airquality))

# odd rows
airquality[odds, ]

# even rows (logical negation)
airquality[!odds, ]
```

Bracket Notation: retrieving columns



opening bracket

closing bracket

```
df[ ,j]
```

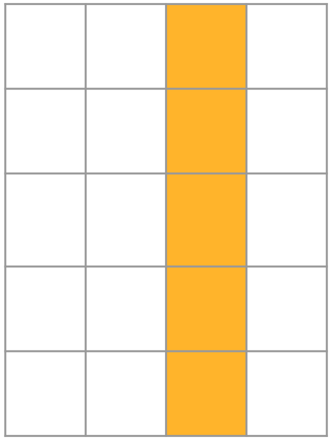
data frame object

empty row index

column index

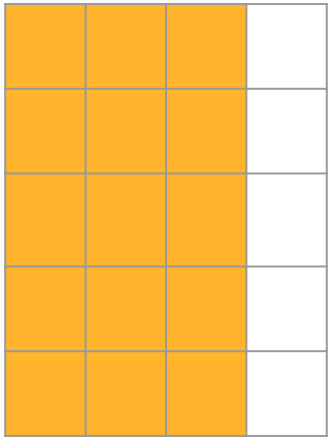
Retrieving Columns

```
df[ ,3]
```



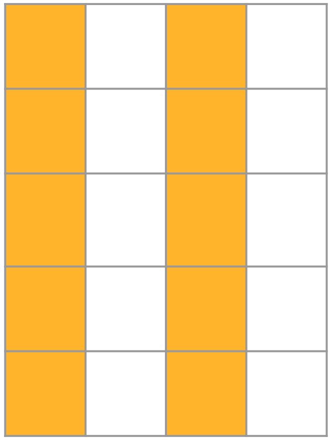
one single column

```
df[ ,1:3]
```



consecutive columns

```
df[ ,c(1,3)]
```



separate columns

Retrieving Columns

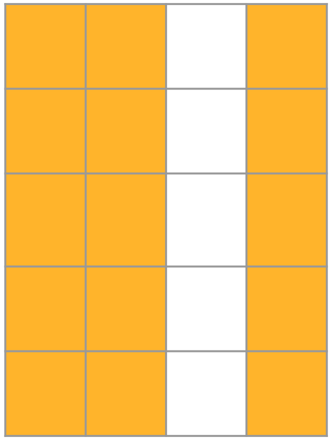
```
# first column
airquality[ ,1]

# columns from 1 to 3
airquality[ ,1:3]

# columns 2, 4, 6
airquality[ ,c(2,4,6)]
```

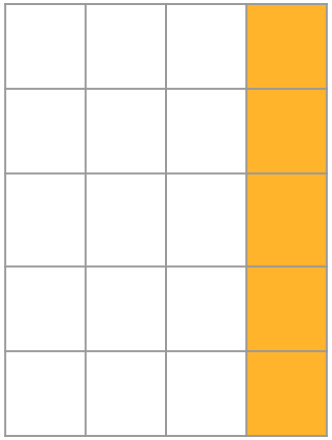
Retrieving Columns (excluding indices)

`df[ ,-3]`



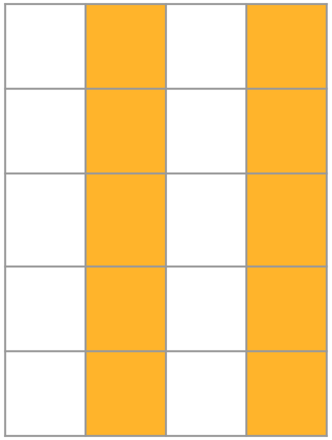
one single
column

`df[ ,-(1:3)]`



consecutive
columns

`df[ ,-c(1,3)]`



separate
columns

Retrieving Columns (excluding indices)

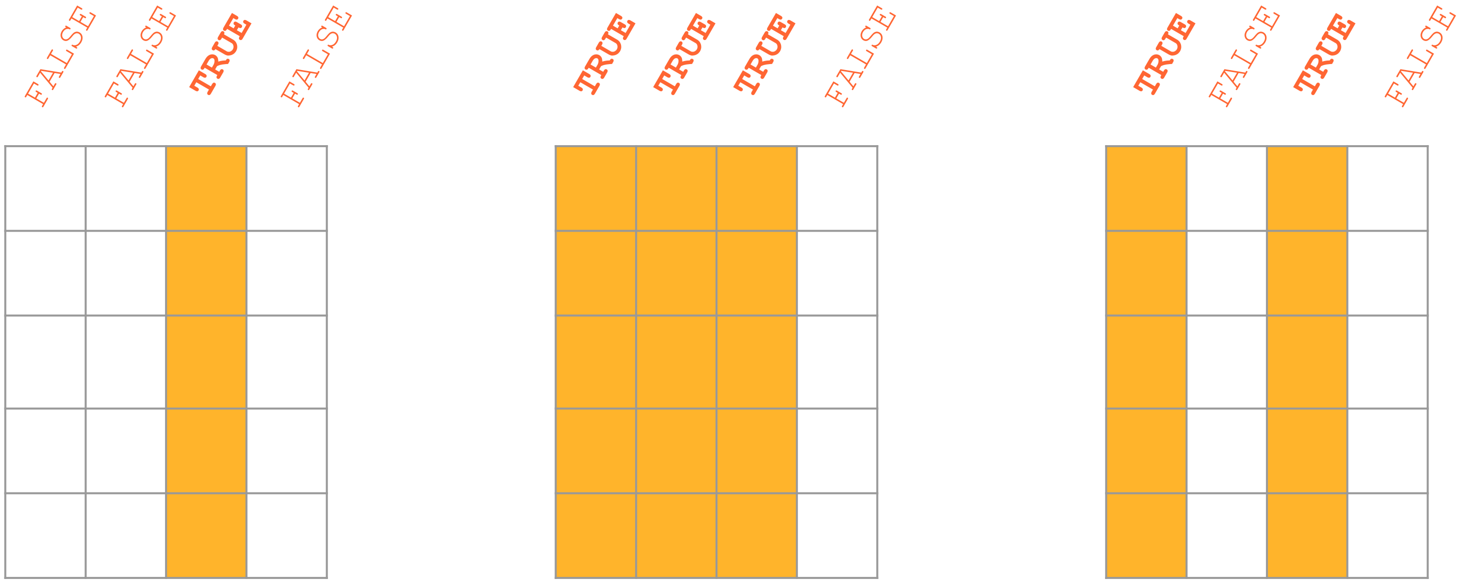
```
# excluding first column
airquality[ ,-1]

# columns except 1 to 3
airquality[ ,-(1:3)]

# all columns but 2, 4, 6
airquality[ ,-c(2,4,6)]
```

Accessing Columns via Logical Subscripts

```
df[ ,logical]
```



Retrieving Columns (logical indexing)

```
# look for these names
these = c('Day', 'Wind', 'Rain',
       'Temp', 'XY', 'Snow')

# query logical selection
Q = names(airquality) %in% these

# selecting corresponding columns
airquality[ ,Q]
```

Retrieving Columns (logical indexing)

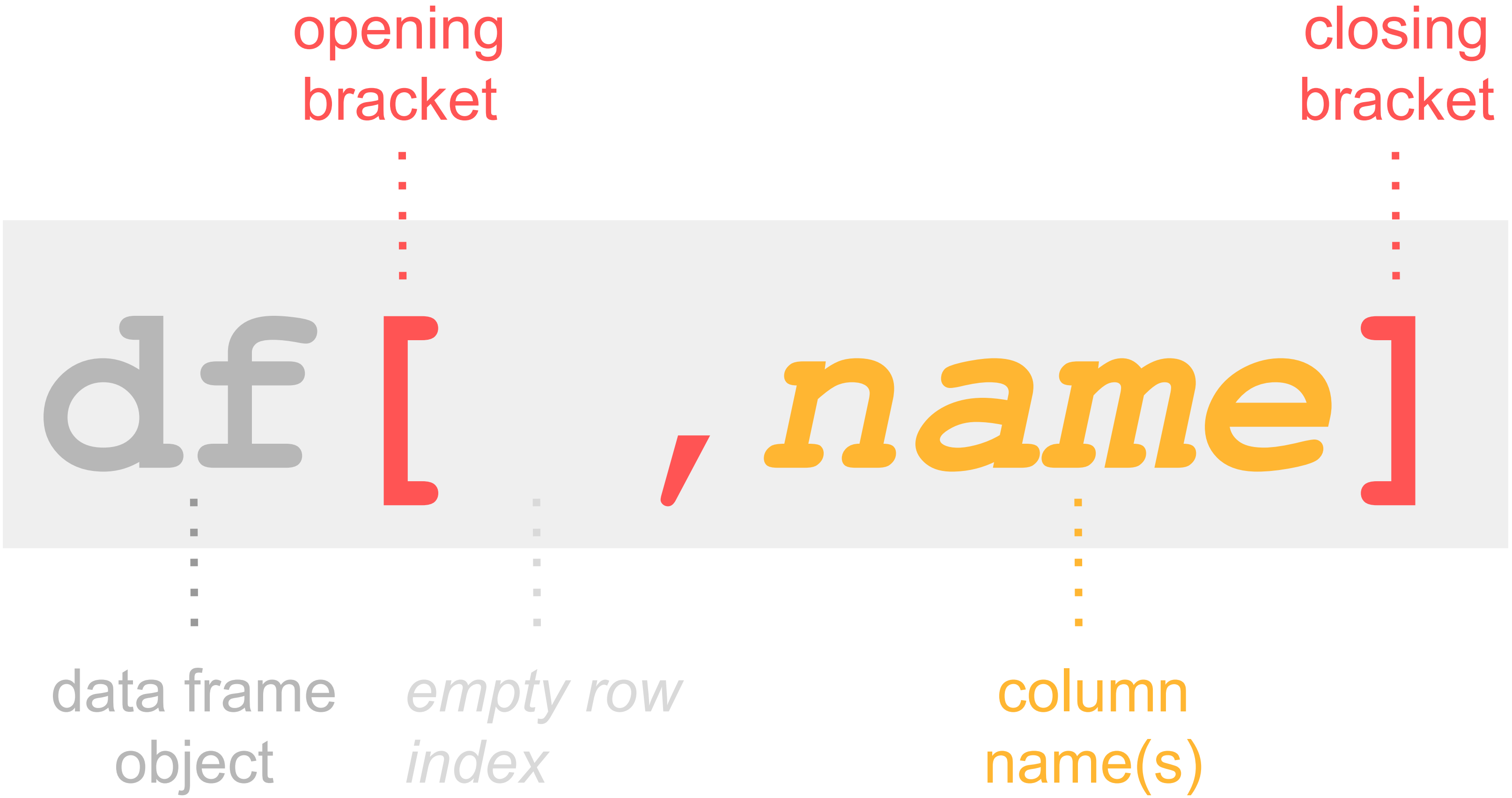
```
# logical vector
cols3 = c(rep(TRUE, 3),
          rep(FALSE, 3))

# first 3 columns
airquality[ ,cols3]

# last 3 columns (logical neg)
airquality[ ,!cols3]
```

More options to access columns

Bracket Notation: retrieving columns via names



opening bracket

closing bracket

```
df[ ,name]
```

data frame object

empty row index

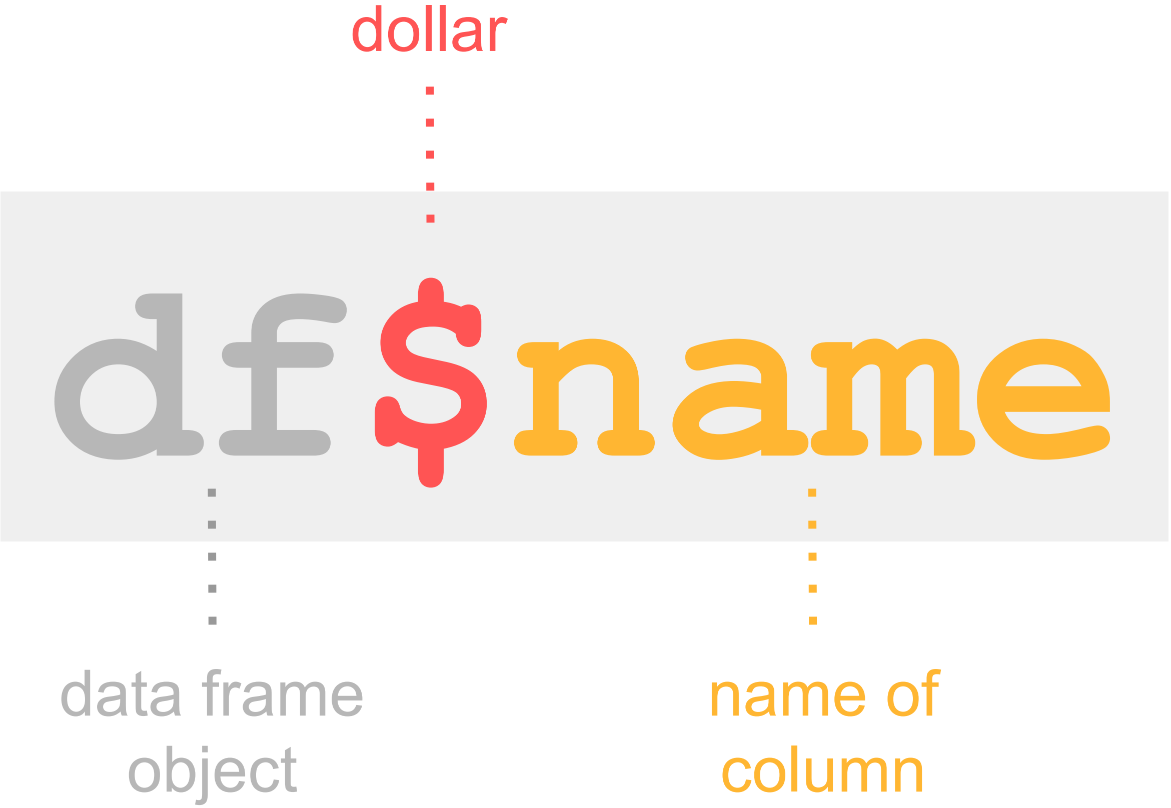
column name(s)

Retrieving Columns (using names)

```
# column Ozone
airquality[ ,"Ozone"]

# columns Wind and Temp
airquality[ ,c("Wind","Temp")]
```

Dollar Notation: retrieving columns via names



dollar

```
df$name
```

data frame object

name of column

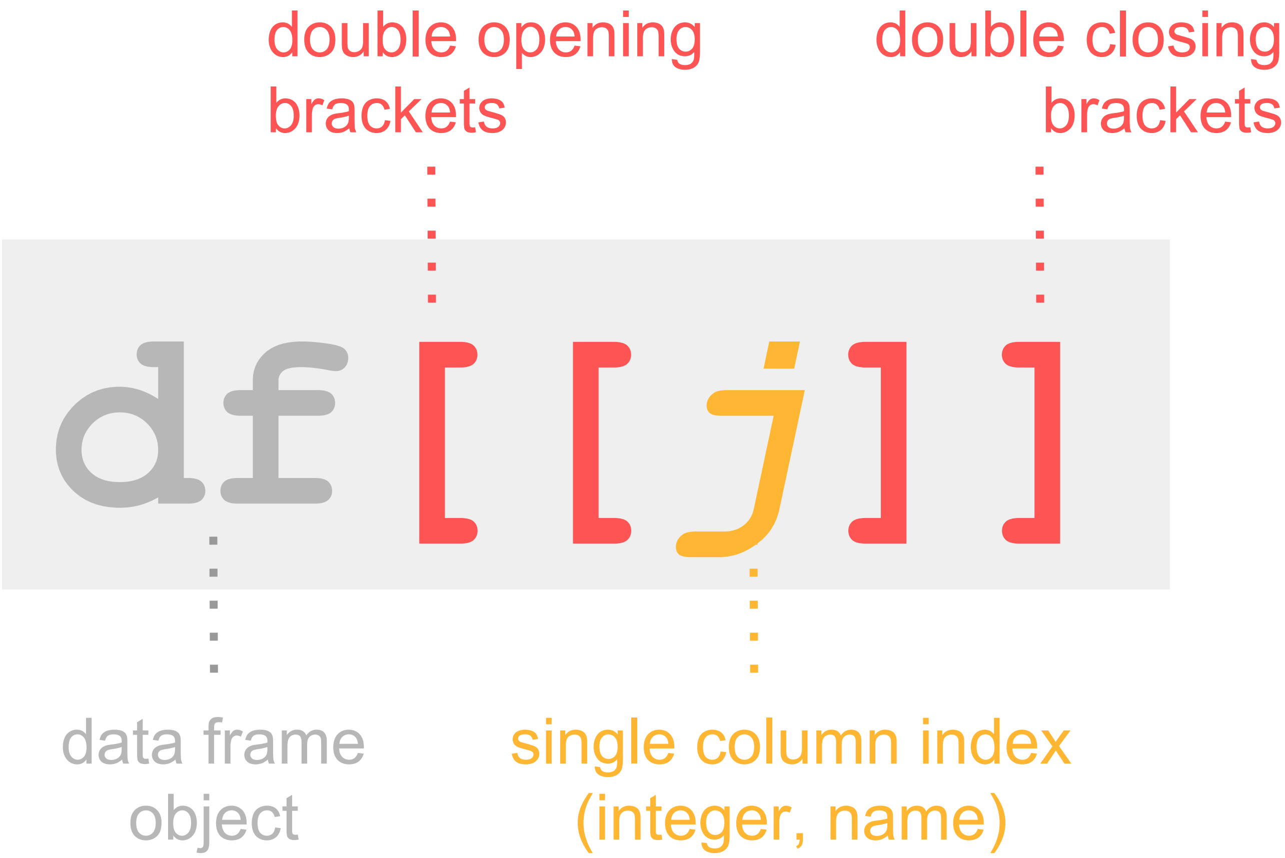
Accessing One Column

```
# column Ozone
airquality$Ozone

# equivalently
airquality$"Ozone"

# equivalently
airquality$'Ozone'
```

Selecting columns with double brackets



double opening brackets

double closing brackets

```
df[[j]]
```

data frame object

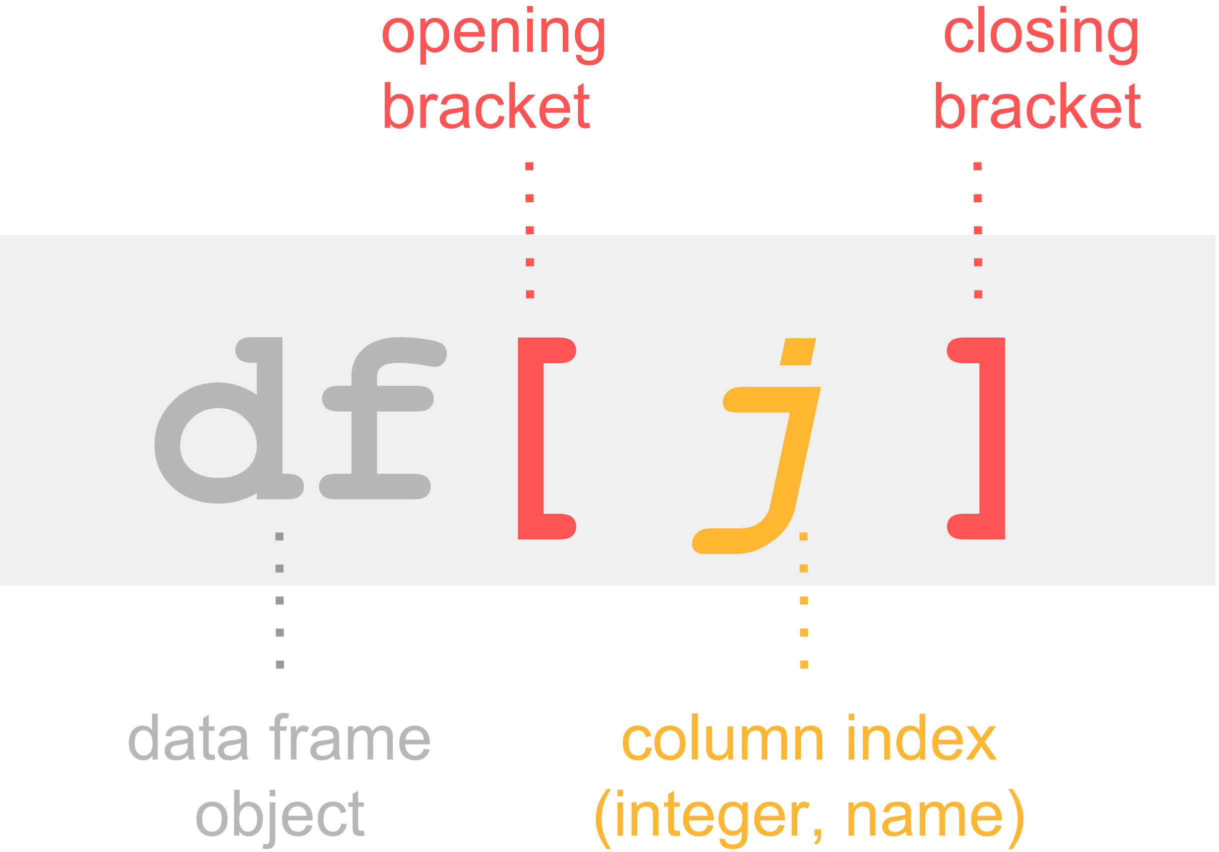
single column index (integer, name)

Accessing One Column

```
# first column
airquality[[1]]

# column Wind
airquality[["Wind"]]
```

Selecting columns with vector notation



opening bracket

closing bracket

```
df[ j ]
```

data frame object

column index (integer, name)

Accessing Columns with vector notation

```
# first column
airquality[1]

# columns from 1 to 3
airquality[1:3]

# columns 2, 4, 6
airquality[c(2,4,6)]
```

Be careful when using this type of syntax since it may create confusion for other users reading your code

Accessing Columns with `list` syntax

```
# column Ozone
airquality["Ozone"]

# columns Ozone and Wind
airquality[c("Ozone","Wind")]
```

Be careful when using this type of syntax since it may create confusion for other users reading your code

Argument `drop` when selecting one column

```
             TRUE
df[i,j,drop=       ]
             FALSE
```

drop

TRUE (default) returns result into a vector

FALSE keeps values as a column

Use **drop** to keep result as a column

```
# first column
airquality[ ,1,drop=FALSE]

# column Ozone
airquality[ ,"Ozone",drop=FALSE]
```