oceanumeric.github.io

```
# create a data.table
dt <- data.table(
                  vn1 = c(1, 100, -567),
                  vn2 = c("hello", "hello","hello"),
                  vn3 = rep("world", 3)
                 )
```

```
# read a csv file into data.table
dt<- fread("file_name.csv")
```

dt[i, j, by]

any operation on columns takes place at j

any operation on rows takes place at i

dt

variable_name_1	variable_name_2	variable_name_3	variable_name_4	vn5	vn7	vn8
integer	numeric (dbl)	character	factor	logic	mixed with missing values	Date/Time
1	2.0	A	female / 1	TRUE	2.0	2017-09-16
100	-3.1415926	"hello"	male / 2	FALSE	"abc"	16:23:57
-567	100	hello world	any categorical data	TRUE	NA	2 June 2020

common functions:

```
str(dt)
summary(dt)
names(dt)
dim(dt)
```

```
#save a data.table into a csv file
fwrite(dt, "file_name.csv")
```

Import key packages

```
library(data.table)
library(magrittr)
library(knitr)
library(ggplot2)
```

Structure of the dataset

```
str(dt)
head(dt)
summary(dt)
names(dt)
setnames('old', 'new')
setorder(vn5, vn6)
sapply(dt, function(x) sum(is.na(x))
```

Check unique or duplicated values

```
dt %>%
    unique(by = c("variables")
dt %>%
    .[duplicated(variable)]
# print out all duplicates
dt %>%
    .[duplicated(variablle) | duplicated(variable, fromLast = TRUE)]
```

dt[i, j, by]

```
# common functions in pipe
%>%
with()
kable()
plot()
par(mfrow = c(2, 2))

# ask ChatGPT always
```

dt

variable_name_1	variable_name_2	variable_name_3	variable_name_4	vn5	vn7	vn8
integer	numeric (dbl)	character	factor	logic	mixed with missing	Date/Time
1	2.0	A	female / 1	TRUE	2.0	2013
100	-3.1415926	hello	male / 2	FALSE	"abc"	2022-09-10
-567	100	hello world	any categorical data	TRUE	NA	09:12:37

```
# class
class(variable)
# is function
is.factor()
is.integer()
is.character()
# as function
as.character()
as.integer()
as.POSIXct()
as.factor()
...
```

Manipulate rows with i

```
# extract rows based on index
dt[5:17, ]  # all columns
dt[1:9, 2:4]  # row 1 to 9, column 2 to 4
# subset rows based on conditions
dt %>%
    .[vn2 >= 20]


# logical operators to use in i

>   <   >=   <=   is.na()   !

&   |   %in%   %like%   %between%



# any functions from dplyr
# could be combined within
# the pipe line
```

Manipulate columns with j

```
# extract columns
dt %>%
    .[, .(vn5, vn7)]
dt %>%
    .[, c(2:6)]  # using column index
# extract columns based on names
dt %>%
    .[, .SD, .SDcols = patterns("^q")]
# extract columns based on type
dt %>%
    .[, .SD, .SDcols = is.integer]
# extract and transfom at the same time
dt %>%
    .[, lapply(.SD, tolower), .SDcols = is.character]
# create a new columns on original data
dt %>%
    .[, vn9 := vn2 + 2]
# create or transform columns on original data
# using name vector, .SDcols, and lapply with :=
```

subgroup with by

```
# summarize vn2 by vn4
dt %>%
    .[, .(vn2_mean = mean(vn2), by = vn4]

# one of the most common way to use by
# is that we need do some operation on one
# or several variables based on another
# categorical variables, such as

- dt[, .(c=sum(b)), by = a]
- dt[, .(c=mean(b)), by = .(a, d)]
```

```
# use keyby if you want to
# sort within the group

# special functions that
# could bring magics
.N, .I .SD
```