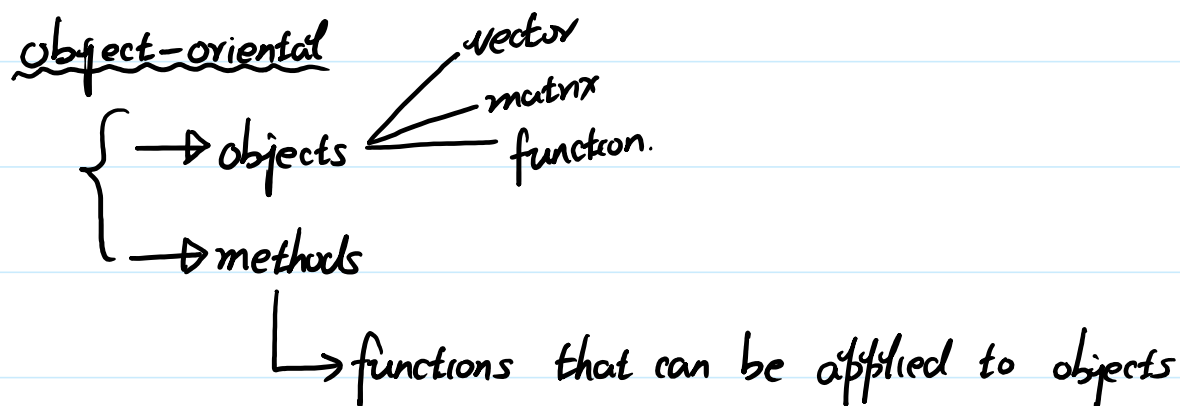


1. Structure:

ex. $X \sim N(0,1) \sim \text{rnorm}()$ or $f, t, \text{chisq}, \text{cauchy}$
 $\text{dnorm}()$
 $\text{pnorm}()$
 $\text{qnorm}()$.

ex. in stata, reg $y \sim x$
 in R, everything is object, $\text{lm}(y \sim x)$
 function.
 object.

ex. $\text{rnorm}()$ is a function.

`help(rnorm)` gives information you need.

`rnorm(n, mean=0, sd=1)`

argument with default value.

`dnorm` has more argument, `log=FALSE`.

saves time computing log-density.

ex. `curve` (`dnorm` (-4, 4))
function function. objective.

ex. `help(plot)` tells you how to graph.

ex. type function, Tab gives all argument.

ex. `y` as time series.

`y <- arima.sim(n=50, model=list(ar=.9))`

To know type of object, use `is(x)`, `is(y)`.

ex. `x <- rnorm(50)`

$\text{is}(x) \Rightarrow \text{"numencat"}, \text{"vector"}$

$\text{is}(y) \Rightarrow \text{"ts"}, \text{"structure"}, \text{"oldclass"}, \text{"vector"}$

Both are collection of elements, but $\text{is}(y)$ has more information.

ex. states, frequency.

Once you have objects, you have methods.

1) Print.

~ type object itself \approx print it.

~ type print gives information.

generic function.

default value \Rightarrow print.default

when calling $\text{print}(x)$,

$\rightarrow \text{print.default}(x)$

but, $\text{print}(y)$,

$\rightarrow \text{print.ts}(y)$

function name of the class.

ex. $y = \beta_0 + \beta_1 z + u.$

$\text{res} \leftarrow \text{lm}(y \sim z).$

↖
Saving regression into object res.

$\text{is}(\text{res}) \Rightarrow \text{"lm", "oldclass"}$

↖
has all information of regression.

$\text{print}(\text{res})$

↳ $\text{print.lm}(\text{res}).$

~ plot is also generic function.

ex. $\text{plot}(x)$ prints dots. $\leadsto \text{plot.default}(x)$

$\text{plot}(y)$ plots time series in lines. $\leadsto \text{plot.ts}(y).$

$\text{plot}(\text{res}) \rightarrow \text{plot.lm}(\text{res}).$

↳ four things.

use $\text{plot}(\text{res}, 4)$ (or 1, 2, 3) produce single graph.

~ summary is also generic function.

ex. $\text{summary}(x) \rightarrow \text{summary.default}(x).$

$\text{summary}(res) \rightarrow \text{summary.lm}(res).$

~ Particular Object.

1) Vector

↳ collection of things.

ex. $x \leftarrow c(1, 2, 3).$ creates column vector $(1, 2, 3)$

$y \leftarrow c(\text{"one"}, \text{"two"}, \text{"three"}).$
character.

$x \leftarrow \text{seq}(1, 100, \text{by}=0.5)$
frequency. or use $\text{length.out}=200$

$x \leftarrow 1:100$ is same as $\text{seq}(1:100).$

Subsetting:

ex. $x[5]$ gives 5th element of x .

$x[c(1, 2, 5, 7)]$ gives 1st, 2nd, 5th, 7th element of x .

or,

$\text{which} \leftarrow c(1, 2, 5, 7)$

$x[\text{which}]$

But,

$x[\text{which} \leftarrow c(1, 2, 5, 7)]$

not same as "=", it saves object to which at same time.

~logical elements

ex. $x < 0$ check < 0 for each element.

`x[x<0]` removes all false elements, print only true ones.

$x[x < 0 \ \& \ x > -1]$ combines logical elements.
or
|

$X \rightarrow \text{Income}$
(100%)

$Y \rightarrow \text{Age.}$
|wx|

Income for individuals older than 40.

$$X_{\text{total}} \leftarrow X[y \geq 40]$$

or, use $X_{old} \leftarrow \text{subset}(x, y \geq 40)$.

$x_{young} \leftarrow x[y < 40]$

or, $x[\text{!}(y \geq 40)]$

not reverse logical output.

2) Matrix

collection of elements

+ attributes 'dim'

$c(\text{row}, \text{col})$

ex. $\text{matrix}(\text{rnorm}(20), 4, 5)$. create 4x5 matrix.

$\text{print}(\text{matrix}) \rightarrow \text{print}(\text{matrix}(\text{matrix}))$

computer does not save matrix.

$\text{attr}(x, "dim")$ gives $c(4, 5)$.

or, use $\text{dim}(x)$.

$\text{ncol}(x)$ and $\text{nrow}(x)$ give # of row/column.

now, you can use subsetting.

$$\dim(x)[1] \rightarrow 4$$

$$\dim(x)[2] \rightarrow 5.$$

$$x[1, 3] \rightarrow 1^{\text{st}} \text{ row, } 3^{\text{rd}} \text{ column.}$$

$$x[c(1, 2), 3] \rightarrow 1^{\text{st}}/2^{\text{nd}} \text{ row.}$$

$$x[1,] \text{ all of } 1^{\text{st}} \text{ row}$$

$$x[, 1] \text{ all of } 1^{\text{st}} \text{ col.}$$