

# **El llenguatge R**



## **Llenguatges de Programació Competència Transversal**

Miquel Gómez i Esteve  
Desembre 2018

## Índex

Introducció .....	3
Paradigmes de Programació .....	4
Sistema de Tipus .....	6
Principals Aplicacions .....	7
Llenguatges similars .....	8
Exemples de Codi .....	9
Bibliografia .....	10

## Introducció

R és un llenguatge de programació Open Source, el qual va ser dissenyat per Ross Ihaka i Robert Gentleman a la Universitat d'Auckland de Nova Zelanda i desenvolupat pel "R Development Core Team". La primera versió va ser publicada ja fa 25 anys, concretament l'any 1993. Aquesta dada dóna a entendre el gran abast del llenguatge, així com la utilitat de les seves funcionalitats i la extensa comunitat. El logo del llenguatge R es mostra a la figura 1.

El llenguatge té moltes funcinalitats, però la gran majoria estàn relacionades amb l'anàlisi, i l'estadística.

Segons l'índex TIOBE publicat el desembre del 2018, el qual mesura la popularitat dels llenguatges de programació, es troba a la posició 16. A la figura 1 situada a continuació, es mostren les primeres posicions del ranking. Tenint en compte aquesta popularitat, el resultat es una gran comunitat i suport, així com un una gran quantitat de llibreries i packages.

Dec 2018	Dec 2017	Change	Programming Language	Ratings	Change
1	1		Java	15.932%	+2.66%
2	2		C	14.282%	+4.12%
3	4	▲	Python	8.376%	+4.60%
4	3	▼	C++	7.562%	+2.84%
5	7	▲	Visual Basic .NET	7.127%	+4.66%
6	5	▼	C#	3.455%	+0.63%
7	6	▼	JavaScript	3.063%	+0.59%
8	9	▲	PHP	2.442%	+0.85%
9	-	▲	SQL	2.184%	+2.18%
10	12	▲	Objective-C	1.477%	-0.02%
11	16	▲	Delphi/Object Pascal	1.396%	+0.00%
12	13	▲	Assembly language	1.371%	-0.10%
13	10	▼	MATLAB	1.283%	-0.29%
14	11	▼	Swift	1.220%	-0.35%
15	17	▲	Go	1.189%	-0.20%
16	8	▼	R	1.111%	-0.80%



**Figura 1:** Logo del llenguatge R.

**Figura 2:** Index TIOBE.

El llenguatge està disponible per als sistemes operatius més utilitzats com són Windows, Mac i Linux.

Es tracta d'un llenguatge interpretat, és a dir, que aquest no necessita d'un prèvi procés de compilació abans de començar a executar-lo i per tant normalment s'utilitza amb un interfície de línia d'ordres (CLI o command line interpreter en anglès), el qual interpreta i avalua línia a línia l'entrada de l'usuari. Cal destacar la interfície gràfica anomenada RStudio la més usada en aquest àmbit.

## Paradigmes de Programació

R és un llenguatge molt extens, amb moltes funcionalitats i llibreries orientades a diferents àmbits, és per aquesta raó que el llenguatge segueix molt paradigmes de programació diferents. A continuació s'en destaquen els més importants que segueix R:

### Imperatiu

Els llenguatges que segueixen aquest paradigma són aquells en els quals l'usuari defineix com l'estat del programa varia mitjançant un seguit de declaracions.

### Orientació a Objectes

En aquest paradigma els llenguatges organitzen les dades en atributs i el codi en funcions / mètodes, propis de cada objecte. Permet que existeixi en el llenguatge el concepte de polimorfisme.

### Array

El paradigma de programació array, permet al llenguatge aplicar una funció o mètode a una estructura de dades gran, per exemple un vector o una matriu, i aquesta aplicació implica realment una operació més petita sobre tots i cada un dels membres de l'estructura inicial. A continuació es mostra un exemple molt bàsic, que consisteix en la declaració de dues matrius i la posterior multiplicació entre elles.

```
> A <- matrix(1:6, nrow=2)
> B <- t( matrix(6:1, nrow=2) )
> C <- A %*% B
> C
      [,1] [,2]
[1,]  28  19
[2,]  40  28
```

**Figura 3:**  
Exemple de programació array a R.

### Procedural

El nom procedural prové de la paraula anglesa procedure, que seria equivalent a dir mètode. Aquest paradigma doncs, permet estructurar i organitzar el codi en rutines, subrutines i funcions, de manera que augmenta la modularitat d'aquest.

### Declaratiu

En un llenguatge declaratiu, a diferència de l'imperatiu, no es defineixen els canvis de l'estat del problema, sinó que es defineix com ha de ser la solució final i el programa és l'encarregat de trobar-la.

Tot i que per definició es podria considerar oposada al paradigma de programació imperatiu, el llenguatge R els compleix tots 2.

## Funcional

Els llenguatges funcionals són aquells que tendeixen a buscar la solució fent ús de diverses funcions matemàtiques. Un altre tret molt característic és la possibilitat d'utilitzar el concepte de funcions de primer ordre, el qual permet tractar les funcions com un objecte.

## Reflectiu

Un llenguatge que segueix el paradigma reflectiu, s'entén com un llenguatge capaç de modificar la seva estructura i el seu propi comportament durant l'execució. A la figura 4 hi ha un petit codi que mostra la crida a la funció hello d'un objecte de tipus foo, en el primer cas sense ser reflectiu i en el segon sí.

**Figura 4:**  
Exemple de crida d'un mètode propi d'un objecte amb i sense reflexió.

```
# Cas 1
obj <- foo()
hello(obj)

# Cas 2
the.class <- "foo"
the.method <- "hello"
obj <- do.call(the.class, list())
do.call(the.method, alist(obj))
```

## Sistema de Tipus

El llenguatge R és un llenguatge amb un sistema de tipus dinàmic, com a conseqüència els tipus de les diferents variables es coneixen en temps d'execució. També el sistema de tipus és inferit, és a dir no cal escriure de manera explícita el tipus de la variable, sinó que el llenguatge infereix el tipus en funció del valor que pren.

Com s'escenifica a la figura 5, el llenguatge R no es caracteritza per tenir gaire seguretat de tipus, ja que una mateixa variable pren un tipus diferent després de cada línia i observem que aquest exemple no ens dóna cap error.

# Entrada	# Sortida
a <- TRUE	[1] TRUE
a	[1] 1
a[1] <- 1	[1] "hola"
a	
a <- "hola"	
a	

**Figura 5:**  
Escenificació de l'absència de seguretat de tipus a R.

Tot i no haver-hi gaire informació al respecte, després de l'exemple previ i altres comprovacions a part, podem dir que el llenguatge R té un sistema de tipus feble (weak typing), ja que no té pràcticament regles de tipus i en temps d'execució el llenguatge fa conversions de tipus implícites.

R suporta molts tipus diferents, per començar tots els tipus que podríem anomenar bàsics, com serien tots els tipus numèrics, strings i caràcters, etc.

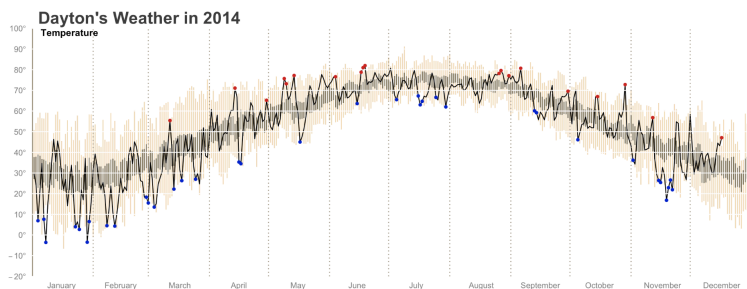
El tipus i estructures més complexes i interessants es comenten a continuació:

- Vector
- Matriu
- Array
- Data frames: Consisteix bàsicament en una matriu on les columnes poden tenir tipus diferents.
- Llistes
- Factors: Són una manera de representar els Data frames, on cada possible valor que hi ha a la columna serà substituït per un valor numèric diferent.
- Funcions: Com s'ha comentat anteriorment el llenguatge R permet tractar funcions com a objectes.

## Principals Aplicacions

R té diferents aplicacions, la gran majoria d'elles relacionades amb càlculs matemàtics, anàlisi estadístic i generació de gràfics. Aquests són alguns dels àmbits en els que s'utilitza R, entre molts d'altres:

- Finances, negocis i economia: R és una eina molt útil de cara a estudiar el mercat i les seves oscil·lacions.
- Medicina biologia i genètica: En el desenvolupament de nous medicaments, R es pot utilitzar per tractar i analitzar les dades experimentals obtingudes. També es pot utilitzar en estudis sobre el genoma.
- Meteorologia: R pot ajudar a estudiar i fer prediccions també en l'àmbit de la meteorologia. A la figura 6 es mostra un gràfic que il·lustra les tendències de temperatura durant un any en una zona concreta.
- Big Data: R té eines disponibles per al tractament de dades massives.
- Machine Learning: També hi ha llibreries de R disponibles que permeten aplicar tècniques pròpies de Machine Learning



**Figura 6:**  
Exemple d'ús de R en un estudi Meteorològic.

Moltes empreses amb molt de renom fan ús del llenguatge R. Algunes d'elles es mostren a continuació:

- Facebook: Utilitza R per a analitzar el comportament dels usuaris sobretot respecte a les actualitzacions de l'estat.
- Google: Fa ús del llenguatge per a evaluar i millorar les campanyes de màrqueting, així com per analitzar el present i futur econòmic de l'empresa.
- Microsoft: L'empresa va decidir comprar el llenguatge i distribuir-ne una nova versió. A part, com a empresa, utilitzen el llenguatge per a diverses funcions.
- Airbnb: Per a tractar ciències de dades a escala fan ús de R.

També podríem incloure en el grup d'empreses que utilitzen R a Twitter, Uber, IBM, Hp, Ford, etc.

La gran quantitat de llibreries i packages, juntament amb les funcionalitats que aquests aporten, permeten que R sigui utilitzat en tots aquests àmbits.

## Llenguatges Similars

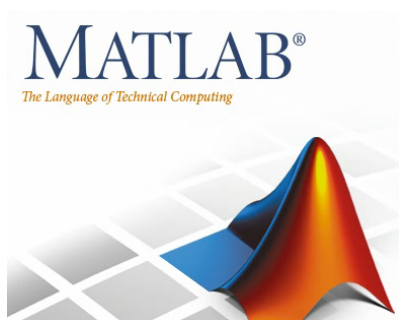
Com s'ha comentat a la introducció, el llenguatge R va ser desenvolupat per el "R Development Core Team" els responsables també del llenguatge S. El llenguatge R, en els seus inicis, va néixer com una branca experimental de S, la qual, amb el temps, va acabar sent un llenguatge per sí sola. Es per aquesta raó que R té moltes similituds amb S, i que una gran quantitat de codi escrit en S pot córrer sense problema en R.

Julia és un llenguatge de scripting que va influenciar molt en el desenvolupament de R, actualment és pot utilitzar R desde Julia fàcilment i comparteixen algunes funcionalitats.

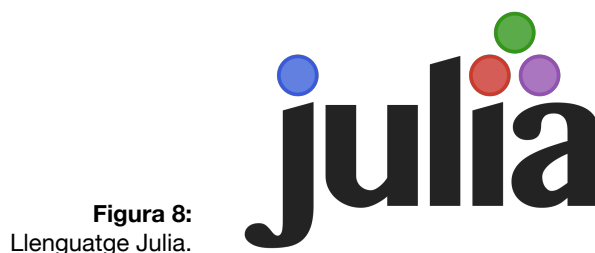
Per les funcionalitats i aplicacions que té R, aquest s'assembla molt a alguns altres, entre els quals he destacat els següents:

- Matlab
- APL (A Programming Language)
- Sage
- Python
- GNU Octave

A les figures 7, 8 i 9 es mostren els logos/imatges que identifiquen alguns dels llenguatges esmentats



**Figura 7:** Matlab.



**Figura 8:**  
Llenguatge Julia.



**Figura 9:** Logo Llenguatge Sage.



## Exemples de Codi

A continuació hi ha uns exemple de codi i una breu explicació.

### Hello World

El mític Hello World, en R es fa ús de la funció print

```
> print("Hello World!")
```

### Conversor de decimal a binari

En aquest exemple es pot veure com es defineix una funció, en aquest cas amb 1 sol paràmetre. La funció és molt simple, només utilitza la condició if per a diferenciar els 2 casos possibles. La funció cat es semblant a print, amb la diferència de que el cat no acaba amb una nova línia.

A continuació es pot veure com es crida aquesta funció des de l'interpret de comandes.

```
convert_to_binary <- function(n) {  
  if(n > 1)  
    convert_to_binary(as.integer(n/2))  
  cat(n %% 2)  
}  
  
> convert_to_binary(133)  
10000101
```

### Llegir input de l'usuari

En aquest exemple es mostra una manera de rebre un input desde l'usuari. En aquest cas, es tracta d'un sol enter. Com a l'exemple anterior, en primer lloc es defineix una funció. La funció readline ens permet rebre l'entrada. A continuació veiem la crida desde l'interpret d'ordres per a que directament ens mostri al mateixa entrada.

```
readinteger <- function()  
{  
  n <- readline(prompt="Enter an integer: ")  
  return(as.integer(n))  
}  
  
> print(readinteger())
```

## Bibliografia

- [https://en.wikipedia.org/wiki/R\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/R_(programming_language))
- <https://www.r-project.org/about.html>
- <https://adv-r.hadley.nz/>
- [https://en.wikipedia.org/wiki/TIOBE\\_index](https://en.wikipedia.org/wiki/TIOBE_index)
- <https://www.tiobe.com/tiobe-index/>
- [https://en.wikipedia.org/wiki/Programming\\_paradigm](https://en.wikipedia.org/wiki/Programming_paradigm)
- <https://www.statmethods.net/index.html>
- [https://en.wikipedia.org/wiki/Type\\_system](https://en.wikipedia.org/wiki/Type_system)
- <https://blog.revolutionanalytics.com/2012/06/the-influences-that-shaped-r-inferno-ish-r.html>
- <https://www.r-bloggers.com/practicing-static-typing-in-r-prime-directive-on-trusting-our-functions-with-object-oriented-programming/>
- [https://rpubs.com/bradleyboehmke/weather\\_graphic](https://rpubs.com/bradleyboehmke/weather_graphic)
- <https://www.r-bloggers.com/companies-using-r/>
- <http://www.rexamples.com/>