



Taller virtual corto  
Short virtual workshop  
Lima, 2-3 septiembre 2021

# HidRología con R HydRology with R

Pedro Rau, PhD  
Hidrólogo



ESCUOLA DE  
POSGRADO  
UTEC



✓ **Instructor:**

**Pedro Rau**

Profesor full-time e investigador Universidad de Ingeniería y Tecnología UTEC - Dpto. Ing. Ambiental. Centro de Investigación y Tecnología del Agua CITA. Hidrólogo investigador en redes internacionales: FR, EC, US, UK ... Consultor en Hidrología y Recursos Hídricos

- ✓ **E.mail:** [pedro.rau.ing@gmail.com](mailto:pedro.rau.ing@gmail.com)
- ✓ **Sitio web:** <http://pedrorau.blogspot.com>
- ✓ **Repositorio del curso:** [http://github.com/hydrocodes/HidRologia\\_corto](http://github.com/hydrocodes/HidRologia_corto)
- ✓ **Requisitos:** Conocimiento básicos en Hidrología y Estadística

**Contenido  
del taller**

1. Introducción a las series de tiempo hidrológicas
2. Entorno R, Rstudio y la nube
3. Análisis exploratorio de datos
4. Introducción a la hidrología en la nube

**Metodología  
« Hands-on »**

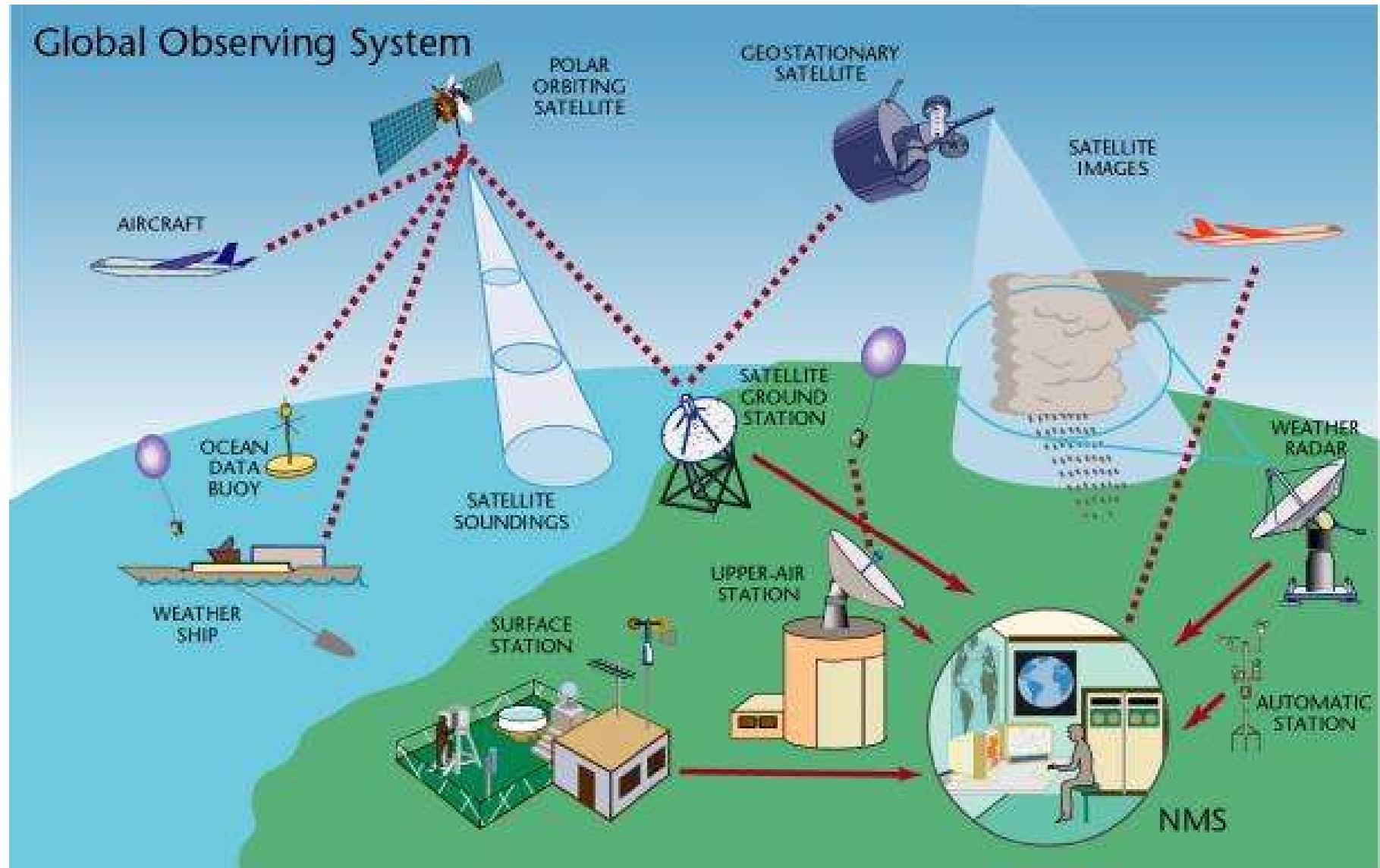
- Lecturas base y planteamiento de ejercicios
- Creación e interpretación de ficheros \*.R con comentarios
- Resolución de bugs

**Total: 4hrs**

# 1. Introducción a las series de tiempo hidrológicas



# 1.1 Sistema de observación global

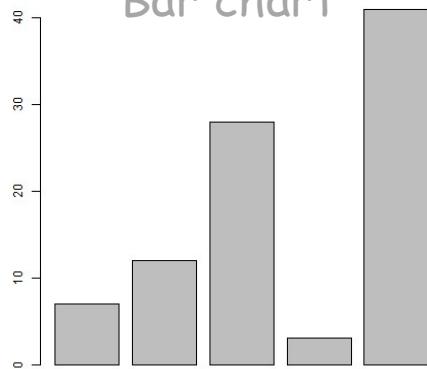


# 1.2 Análisis preliminar de datos hidrológicos

## a. Algunos tipos de representaciones gráficas

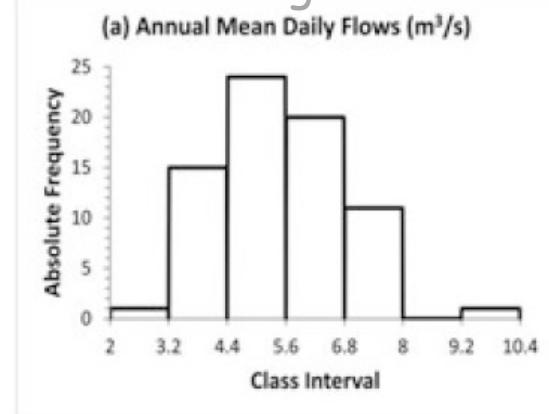
Diagrama de barras

Bar chart



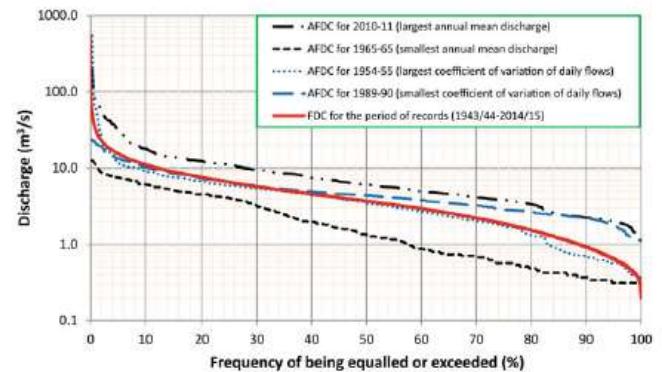
Histograma

Histogram



Curvas de duración

Duration curves



Parámetro de la población

### 1. Punto medio

Media aritmética

$$\mu = E(X) = \int_{-\infty}^{\infty} x f(x) dx \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Mediana

$x$  tal que  $F(x) = 0.5$

Estadística de la muestra

Valor de la información en el 50o. percentil

Coeficiente de variación

$$CV = \frac{\sigma}{\mu} \quad CV = \frac{s}{\bar{x}}$$

### 2. Variabilidad

Varianza

$$\sigma^2 = E[(x - \mu)^2]$$

$$\left( \prod_{i=1}^n x_i \right)^{1/n}$$

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2$$

Desviación estándar

$$\sigma = \{E[(x - \mu)^2]\}^{1/2}$$

$$s = \left[ \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right]^{1/2}$$

### 3. Simetría

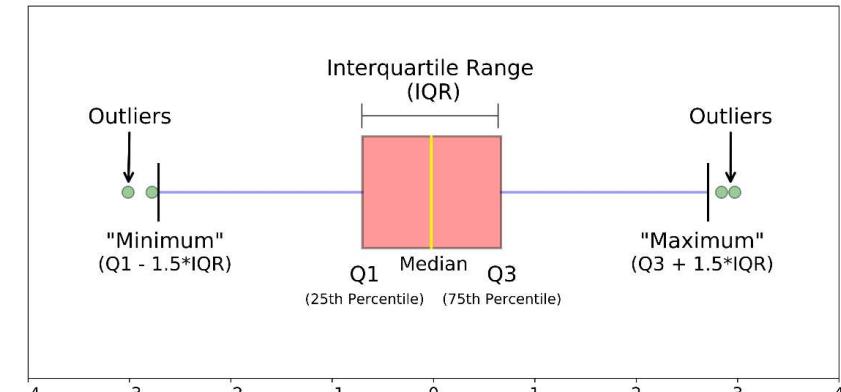
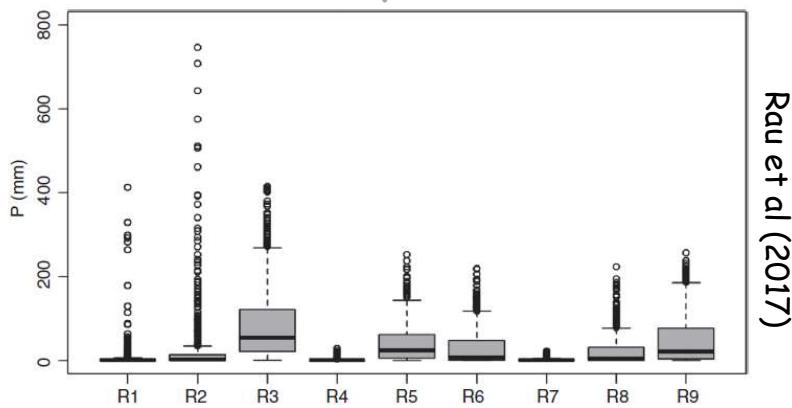
Coeficiente de asimetría (oblicuidad)

$$\gamma = \frac{E[(x - \mu)^3]}{\sigma^3} \quad C_s = \frac{n \sum_{i=1}^n (x_i - \bar{x})^3}{(n-1)(n-2)s^3}$$

## b. Estadísticos descriptivos

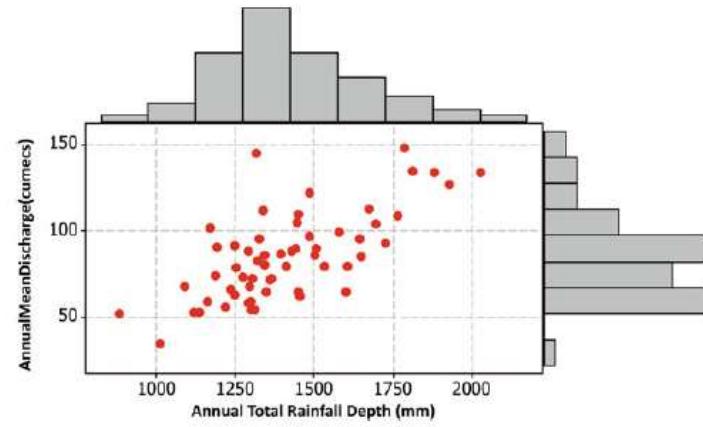
## c. Algunos métodos exploratorios

Diagrama de cajas  
Box plot



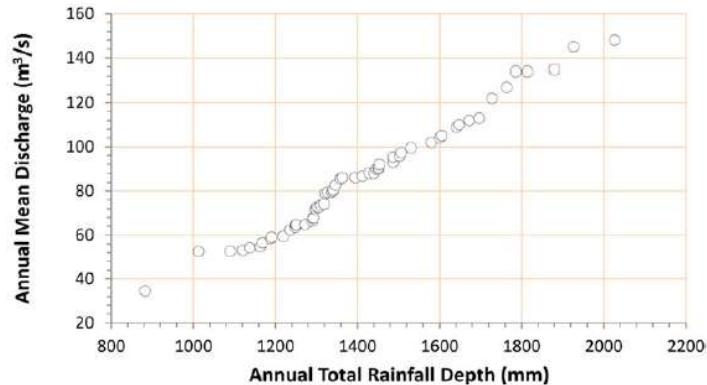
## d. Asociación de datos

Gráfico de dispersion  
Scatter plot



P. Rau –hidRología (2021)

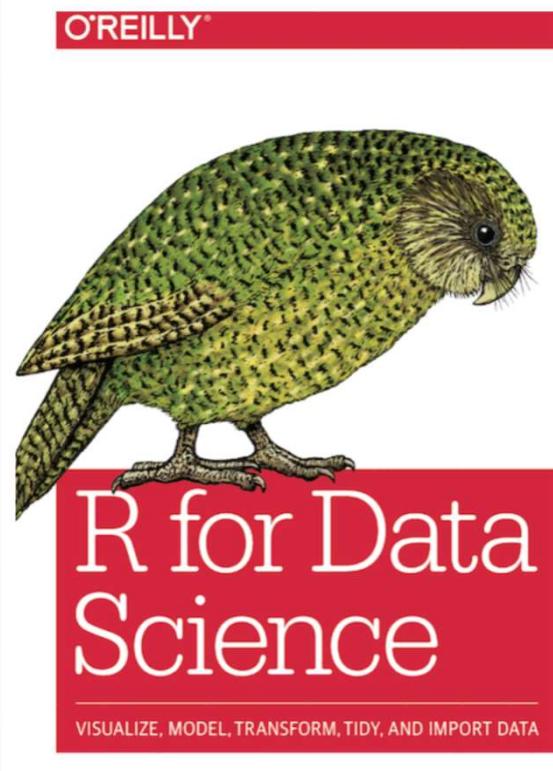
Diagrama Cuantil-Cuantil  
Empirical Quantile-Quantile Diagram  
Q-Q Plot





## 2. Entorno R y Rstudio

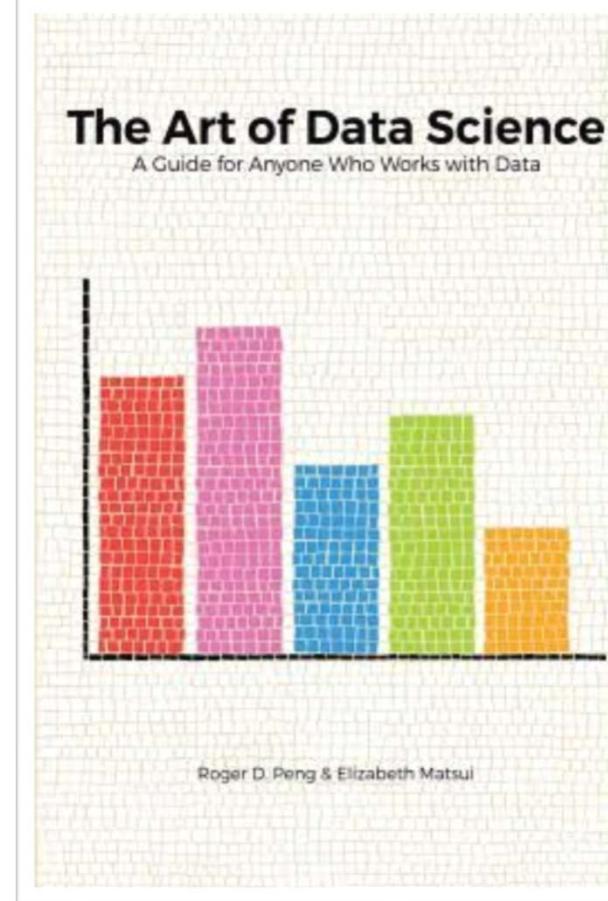




<https://r4ds.had.co.nz/>

<https://es.r4ds.hadley.nz/>

<https://github.com/jrnold/r4ds-exercise-solutions/blob/master/README.md>



<https://bookdown.org/rdpeng/artofdatascience/>

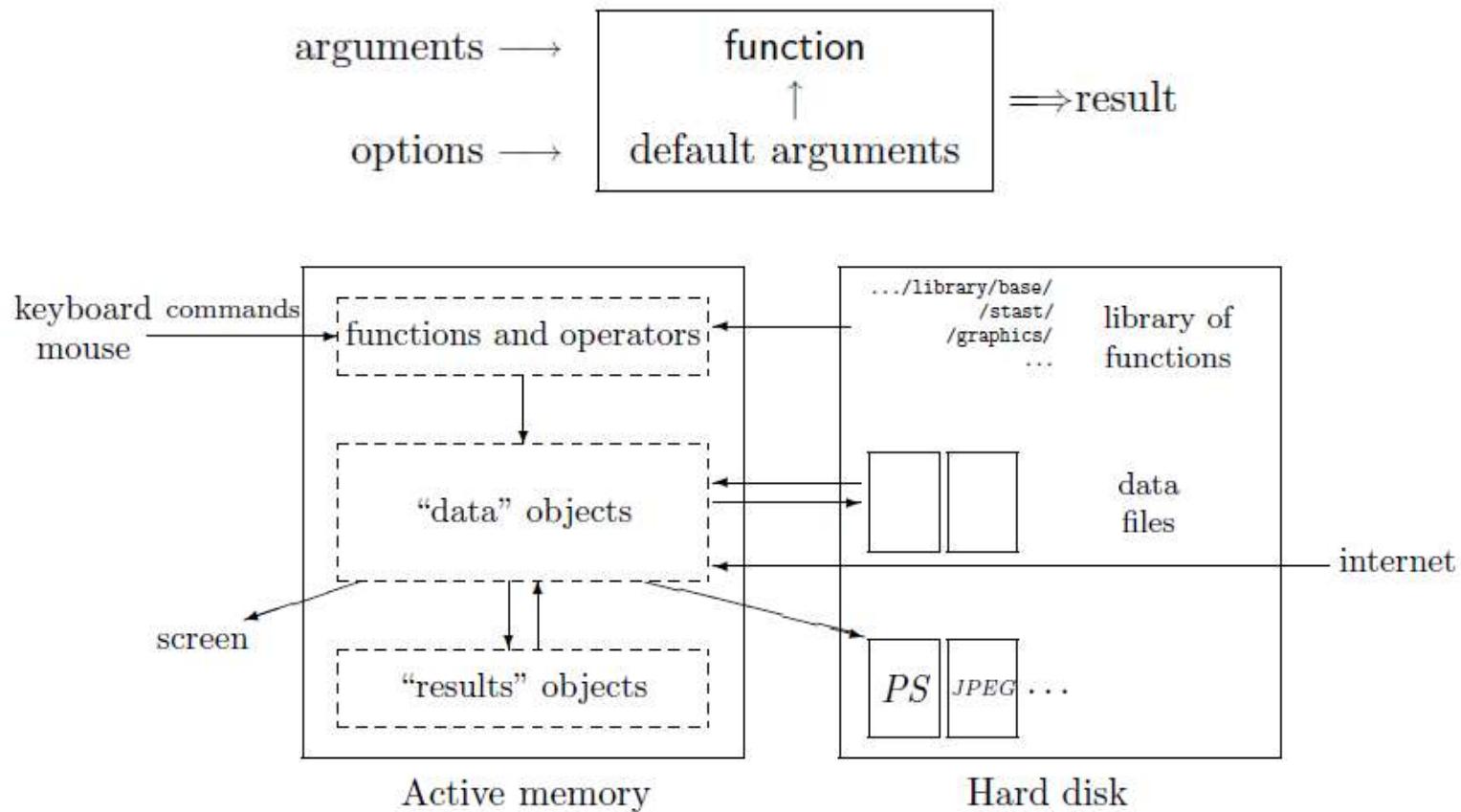
<https://github.com/waldronlab/The-Art-of-Data-Science/blob/master/README.md>



## R-4.1.1 for Windows (32/64 bit) - Agosto 2021

<https://cran.r-project.org/bin/windows/base/>

- ✓ Sistema para el análisis estadístico y graficación (Ihaka y Gentleman, 1996)
- ✓ Lenguaje interpretado, no es un lenguaje compilado
- ✓ Fácil e intuitivo, estrictamente «no es un lenguaje de programación»

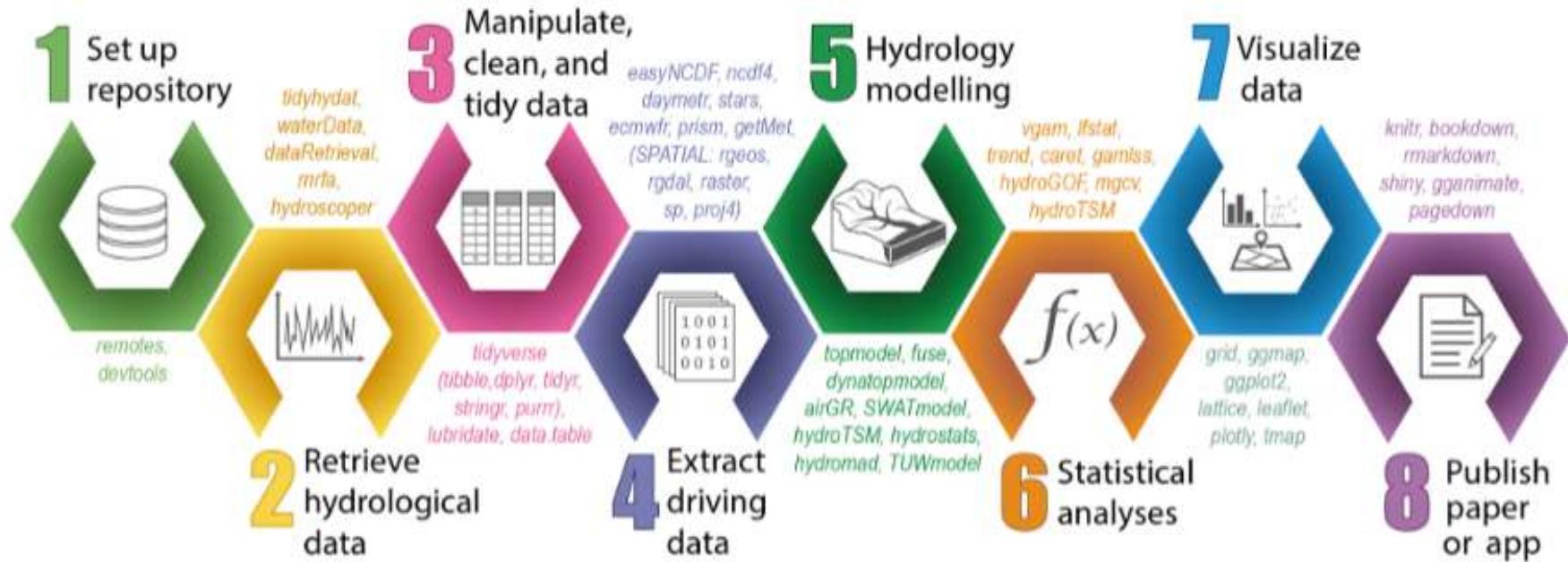


Paradis (2002)

# The Comprehensive R Archive Net-work (CRAN)

<https://cran.r-project.org>

## Tipico Flujo de trabajo en R en hidrologia



Slater et al (2019)

<-

“Clases” de objetos básicos o “atomic” en R:

- numeric (real numbers)
- integer
- complex
- logical (True/False)

( )

## Tipos de objetos en R para representar a los datos

object	modes	several modes possible in the same object?	
vector	numeric, character, complex <i>or</i> logical	No	<code>c()</code>
factor	numeric <i>or</i> character	No	<code>factor()</code>
array	numeric, character, complex <i>or</i> logical	No	<code>array()</code>
matrix	numeric, character, complex <i>or</i> logical	No	<code>matrix()</code>
data frame	numeric, character, complex <i>or</i> logical	Yes	<code>data.frame()</code>
ts	numeric, character, complex <i>or</i> logical	No	<code>ts()</code>
list	numeric, character, complex, logical, function, expression, ...	Yes	<code>list()</code>

#

## Convirtiendo clases de objetos

Conversion to	Function	Rules
numeric	as.numeric	FALSE → 0 TRUE → 1 "1", "2", ... → 1, 2, ... "A", ... → NA
logical	as.logical	0 → FALSE other numbers → TRUE "FALSE", "F" → FALSE "TRUE", "T" → TRUE other characters → NA
character	as.character	1, 2, ... → "1", "2", ... FALSE → "FALSE" TRUE → "TRUE"

## Funciones de probabilidad

Distribución/función	función
Gausse (normal)	rnorm(n, mean=0, sd=1)
exponencial	rexp(n, rate=1)
gamma	rgamma(n, shape, scale=1)
Poisson	rpois(n, lambda)
Weibull	rweibull(n, shape, scale=1)
Cauchy	rcauchy(n, location=0, scale=1)
beta	rbeta(n, shape1, shape2)
'Student' ( <i>t</i> )	rt(n, df)
Fisher-Snedecor ( <i>F</i> )	rf(n, df1, df2)
Pearson ( $\chi^2$ )	rchisq(n, df)
binomial	rbinom(n, size, prob)
geométrica	rgeom(n, prob)
hypergeométrica	rhyper(nn, m, n, k)
logística	rlogis(n, location=0, scale=1)
lognormal	rlnorm(n, meanlog=0, sdlog=1)
binomial negativa	rnbnom(n, size, prob)
uniforme	runif(n, min=0, max=1)
Estadístico de Wilcoxon's	rwilcox(nn, m, n), rsignrank(nn, n)

## Operadores en R

Operators				
Arithmetic	Comparison		Logical	
+	addition	<	lesser than	! x logical NOT
-	subtraction	>	greater than	x & y logical AND
*	multiplication	<=	lesser than or equal to	x && y id.
/	division	>=	greater than or equal to	x   y logical OR
^	power	==	equal	x    y id.
%%	modulo	!=	different	xor(x, y) exclusive OR
%%%	integer division			

<code>plot(x)</code>	plot of the values of <code>x</code> (on the <code>y</code> -axis) ordered on the <code>x</code> -axis
<code>plot(x, y)</code>	bivariate plot of <code>x</code> (on the <code>x</code> -axis) and <code>y</code> (on the <code>y</code> -axis)
<code>sunflowerplot(x, y)</code>	id. but the points with similar coordinates are drawn as a flower which petal number represents the number of points
<code>pie(x)</code>	circular pie-chart
<code>boxplot(x)</code>	"box-and-whiskers" plot
<code>stripchart(x)</code>	plot of the values of <code>x</code> on a line (an alternative to <code>boxplot()</code> for small sample sizes)
<code>coplot(x~y   z)</code>	bivariate plot of <code>x</code> and <code>y</code> for each value (or interval of values) of <code>z</code>
<code>interaction.plot(f1, f2, y)</code>	if <code>f1</code> and <code>f2</code> are factors, plots the means of <code>y</code> (on the <code>y</code> -axis) with respect to the values of <code>f1</code> (on the <code>x</code> -axis) and of <code>f2</code> (different curves); the option <code>fun</code> allows to choose the summary statistic of <code>y</code> (by default <code>fun=mean</code> )
<code>matplot(x,y)</code>	bivariate plot of the first column of <code>x</code> vs. the first one of <code>y</code> , the second one of <code>x</code> vs. the second one of <code>y</code> , etc.
<code>dotchart(x)</code>	if <code>x</code> is a data frame, plots a Cleveland dot plot (stacked plots line-by-line and column-by-column)
<code>fourfoldplot(x)</code>	visualizes, with quarters of circles, the association between two dichotomous variables for different populations ( <code>x</code> must be an array with <code>dim=c(2, 2, k)</code> , or a matrix with <code>dim=c(2, 2)</code> if <code>k = 1</code> )
<code>assocplot(x)</code>	Cohen-Friendly graph showing the deviations from independence of rows and columns in a two dimensional contingency table
<code>mosaicplot(x)</code>	'mosaic' graph of the residuals from a log-linear regression of a contingency table
<code>pairs(x)</code>	if <code>x</code> is a matrix or a data frame, draws all possible bivariate plots between the columns of <code>x</code>
<code>plot.ts(x)</code>	if <code>x</code> is an object of class "ts", plot of <code>x</code> with respect to time, <code>x</code> may be multivariate but the series must have the same frequency and dates
<code>ts.plot(x)</code>	id. but if <code>x</code> is multivariate the series may have different dates and must have the same frequency
<code>hist(x)</code>	histogram of the frequencies of <code>x</code>
<code>barplot(x)</code>	histogram of the values of <code>x</code>
<code>qqnorm(x)</code>	quantiles of <code>x</code> with respect to the values expected under a normal law
<code>qqplot(x, y)</code>	quantiles of <code>y</code> with respect to the quantiles of <code>x</code>
<code>contour(x, y, z)</code>	contour plot (data are interpolated to draw the curves), <code>x</code> and <code>y</code> must be vectors and <code>z</code> must be a matrix so that <code>dim(z)=c(length(x), length(y))</code> ( <code>x</code> and <code>y</code> may be omitted)
<code>filled.contour (x, y, z)</code>	id. but the areas between the contours are coloured, and a legend of the colours is drawn as well
<code>image(x, y, z)</code>	id. but the actual data are represented with colours
<code>persp(x, y, z)</code>	id. but in perspective
<code>stars(x)</code>	if <code>x</code> is a matrix or a data frame, draws a graph with segments or a star where each row of <code>x</code> is represented by a star and the columns are the lengths of the segments
<code>symbols(x, y, ...)</code>	draws, at the coordinates given by <code>x</code> and <code>y</code> , symbols (circles, squares, rectangles, stars, thermometres or "boxplots") which sizes, colours, etc, are specified by supplementary arguments
<code>termplot(mod.obj)</code>	plot of the (partial) effects of a regression model ( <code>mod.obj</code> )

## Ploteando gráficos

`add=FALSE` if `TRUE` superposes the plot on the previous one (if it exists)

`axes=TRUE` if `FALSE` does not draw the axes and the box

`type="p"` species the type of plot, "p": points, "l": lines, "b": points connected by lines, "o": id. but the lines are over the points, "h": vertical lines, "s": steps, the data are represented by the top of the vertical lines, "S": id. But the data are represented by the bottom of the vertical lines

`xlim=`, `ylim=` species the lower and upper limits of the axes, for example  
with `xlim=c(1, 10)` or `xlim=range(x)`

`xlab=`, `ylab=` annotates the axes, must be variables of mode character

`main=` main title, must be a variable of mode character

`sub=` sub-title (written in a smaller font)

# RStudio Desktop 1.4

<https://rstudio.com/products/rstudio/download/#download>



Entorno de desarrollo integrado para el lenguaje de programación R

The screenshot shows the RStudio desktop application interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help, and Addins. The main window has several tabs open: 'cimp5\_extract.R', 'cuena\_pisco.R', 'Heatmap\_Boxplot\_ts.R', and 'script\_m.R'. The 'Environment' pane on the right lists various objects like 'vals', 'Variab', 'Variab\_v', 'Variables', 'VPPgis', 'VPPmonth', 'xorig', 'XParameterer', 'xy', 'AET', 'Convertfun', 'date.ATSM', 'DatesR', 'datos\_vector', 'datos\_num', 'datos\_ts', 'datos\_vector', 'dist', 'i', 'cnr', 'POSIXct', 'Date', 'chr', 'num', and 'Time-Series'. The 'Console' pane at the bottom shows R commands and their results, including a time series plot of 'Tmin (°C/d)' from 2005 to 2030. A red box highlights the 'Editor de código' and 'Consola de scripts' in the code editor area. Another red box highlights the 'Area de objetos e historial' in the Environment pane. A third red box highlights the 'Consola R' and 'Consola de resultados' in the Console pane. A fourth red box highlights the 'Figuras y archivos' in the Plot viewer area.

# Extracción de una variable del CMIP5 (o cualquier archivo .NC)

library(raster)

# Comando brick lee todas las capas contenidas en el archivo nc, Ejm: pr:precipitacion

b <- brick("C:/1\_THESE\_SDUE/3\_SIG/ccmodels/hadgem2-es/tasmin\_Amon\_HadGEM2-ES\_rcp45\_r1i1p1\_20

# Asignando indices

idx <- melt(b)

# Indicar coordenadas de la serie. Coordenadas: longitud y latitud

coord <- c(-70.0, -11.8) # longitud y -11.8 la latitud

vals <- extract(b, coord, df=T)-273.15

# Fijar fechas y datos en un solo archivo dataframe

df <- data.frame(idx, vals[, 2])

rownames(df) <- NULL

names(df) <- c('date','value')

# Dar un vistazo al archivo

head(df)

# Plotear la serie

plot(df, type="l", xlab="Años", ylab="Tmin (°C/d)")

# Fijar fechas y datos en un solo archivo dataframe

df <- data.frame(idx, t(vals)[-1,])

rownames(df) <- NULL

names(df) <- c('date','value')

# Dar un vistazo al archivo

head(df)

# Plotear la serie

plot(df, type="l", xlab="Años", ylab="Tmin (°C/d)")

Tmin (°C/d)

Años

## A. Comandos y códigos en el entorno Rstudio

### Ejercicio A.1

En la consola de resultados, efectuar las siguientes operaciones:

- 1)  $3^2 - 5 * 9 * (25 - 18)$
- 2) Crear una serie consecutiva del 1 al 8
- 3) Crear una serie consecutiva de 8 letras comenzando por "a"
- 4) Asignar la operación  $8 * 5^2$  al objeto value
- 5) Visualizar el objeto value creado
- 6) Crear el objeto p, almacenando una secuencia desde 5 hasta 15 de 2 en 2
- 7) Visualizar p
- 8) Almacenar los siguientes valores de lluvia anual (mm/año) de diferentes estaciones pluviométricas en una cuenca hidrográfica: 200, 210.2, 490, 100.5, 150.1, 190, 310 y 200.2 en el objeto rain.
- 9) Visualizar rain
- 10) Obtener el 5to elemento de rain
- 11) Almacenar los siguientes valores de Elevación (msnm) : 3200, 3500, 4500, 3050, 3100, 2800, 3800, 3500 en el objeto elev.
- 12) Visualizar elev.

# Respuestas A.1

```
> 3^2-5*9*(25-18)
[1] -306
> 1:8
[1] 1 2 3 4 5 6 7 8
> letters[1:8]
[1] "a" "b" "c" "d" "e" "f" "g" "h"
> value<-8*5^2 #Asignamos una operación al objeto value
> value #Escribimos el objeto "value"
[1] 200
> p<-seq(from=5, to=15, by=2) #secuencia desde 5 hasta 15 de 2 en 2
> print (p)
> rain<-c(200, 210.2, 490, 100.5, 150.1, 190, 310, 200.2) #Uso de función
c(combine) asignar datos al objeto rain
> rain #Escribimos el objeto "rain"
[1] 200 210.2 490 100.5 150.1 190 310 200.2
> rain[5] #Obtenemos el 5to elemento del objeto rain
[1] 150.1
> elev<-c(3200, 3500, 4500, 3050, 3100, 2800, 3800, 3500) # Ejemplo de
valores de elevaciones
> elev #Escribimos el objeto "elev"
[1] 3200 3500 4500 3050 3100 2800 3800 3500
```

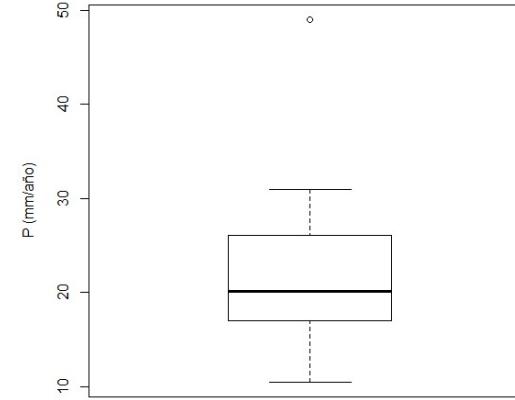
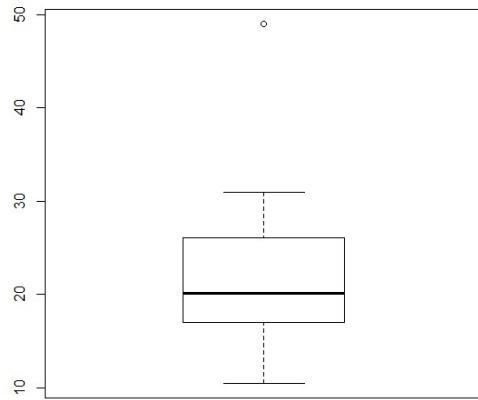
## Ejercicio A.2

En la consola de resultados, efectuar las siguientes operaciones:

- 1) Obtener la media, mediana, desviación estándar, varianza del objeto rain (del ejercicio A.1)
- 2) Visualizar un resumen de los estadísticos notables
- 3) Plotear un diagrama de cajas con los valores del objeto rain. interpretar
- 4) Plotear lo anterior con una etiqueta en el eje vertical indicando: P (mm/año)

# Respuestas A.2

```
> mean(rain)
[1] 23.25
> median(rain)
[1] 20.1
> sd(rain)
[1] 11.91781
> var(rain)
[1] 142.0343
> summary(rain)
   Min. 1st Qu. Median Mean 3rd Qu. Max.
10.50  18.02  20.10  23.25  23.65  49.00
> boxplot(rain) #Grafico de cajas
> boxplot(rain, ylab="P (mm/año)") #Grafico de cajas y etiqueta
```



## Ejercicio A.3

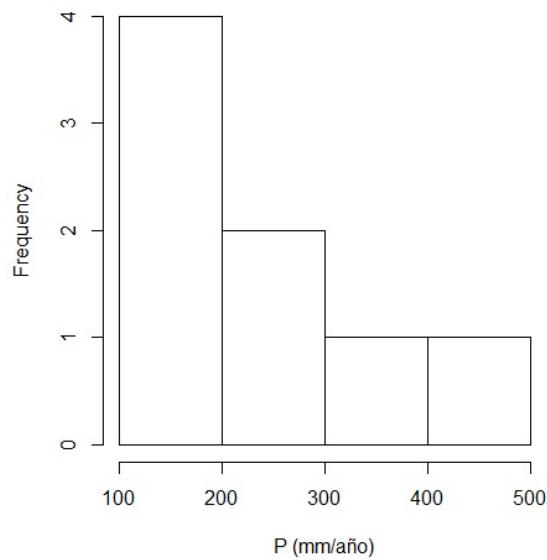
En la consola de resultados, efectuar las siguientes operaciones:

- 1) Plotear un histograma de frecuencias para rain, empleando la frecuencia
- 2) Plotear un histograma de frecuencias para rain, empleando la densidad
- 3) Agregar al histograma una curva de distribución normal en color rojo con la media y desviación estimados anteriormente
- 4) Agregar en un solo grafico (1filas x 2 columnas), el grafico de cajas anterior y el histograma
- 5) Separar los gráficos anteriores
- 6) Calcular la covarianza entre la lluvia y las elevaciones
- 7) Calcular el coeficiente de correlación entre la lluvia y las elevaciones
- 8) Calcular la ecuación de regresión lineal entre la lluvia y la elevación
- 9) Plotear un grafico de dispersión entre la lluvia y la elevación
- 10) Agregar la línea de tendencia al grafico anterior.

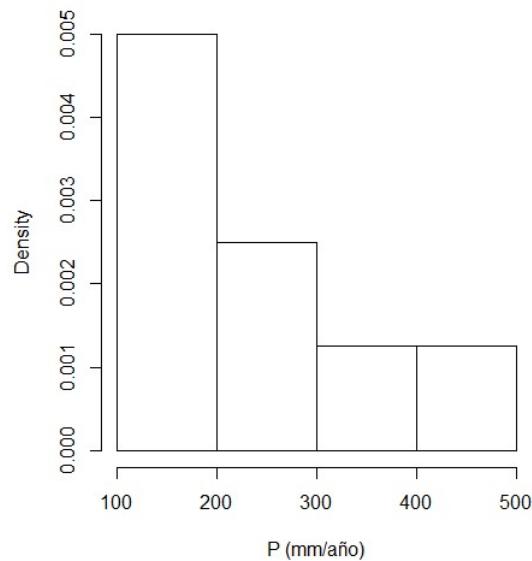
# Respuestas A.3

```
> hist(rain, main="Precipitacion anual", xlab="P (mm/año)", freq=F) # Grafico de histograma, F(False) no usar frecuencia
> curve(dnorm(x, mean(rain, na.rm = T), sd(rain, na.rm = T)), add = TRUE, col="red") # Agregar curva distribucion r normal en color rojo
> split.screen(c(1,2)) #usar este comando para ordenar el espacio de figuras, aquí 2 figuras en una fila
> screen(2) #usar este comando antes de plotear la segunda figura
> close.screen(all=TRUE) # comando para regresar a las condiciones iniciales de una figura por ventana
> cov(rain,elev) #covarianza
[1] 58855.89
> cor(rain,elev) #coeficiente de correlación r
[1] 0.9182557
> lm(elev ~ rain) #ecuación de regresión lineal
Call:
lm(formula = elev ~ rain)
Coefficients:
(Intercept)      rain
2486.688       4.082
> plot(rain,elev,main="Elevation-Rainfall relationship",xlab="Rainfall (mm/yr)",
ylab="Elevation (masl)")
> abline(lm(elev~rain))
```

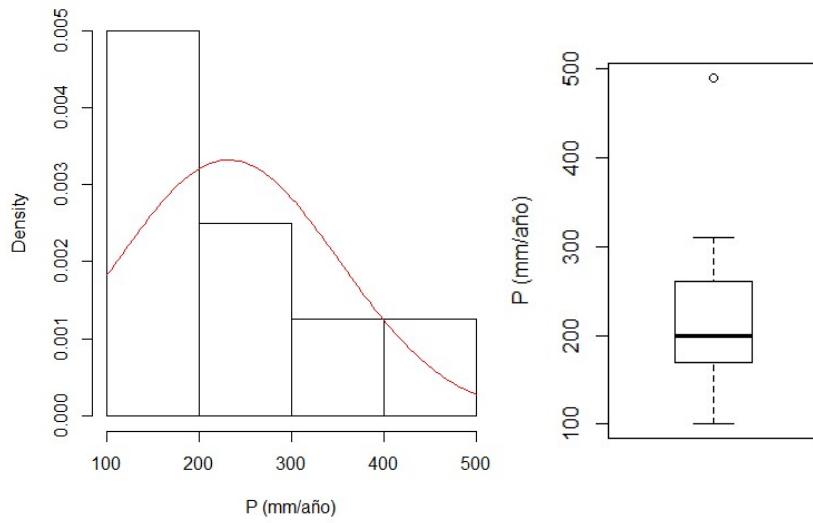
Precipitacion anual



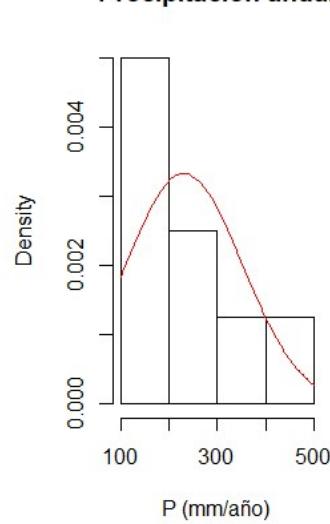
Precipitacion anual



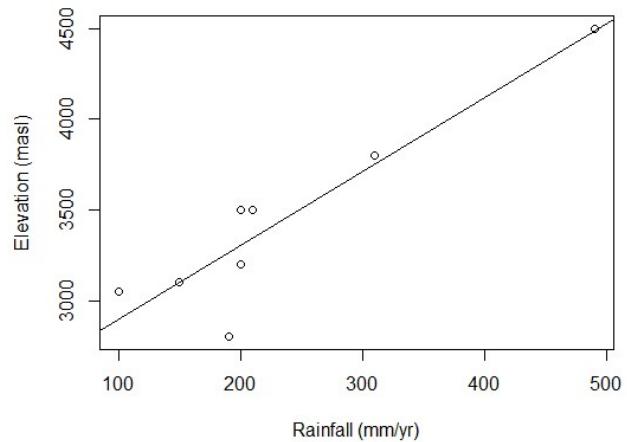
Precipitacion anual



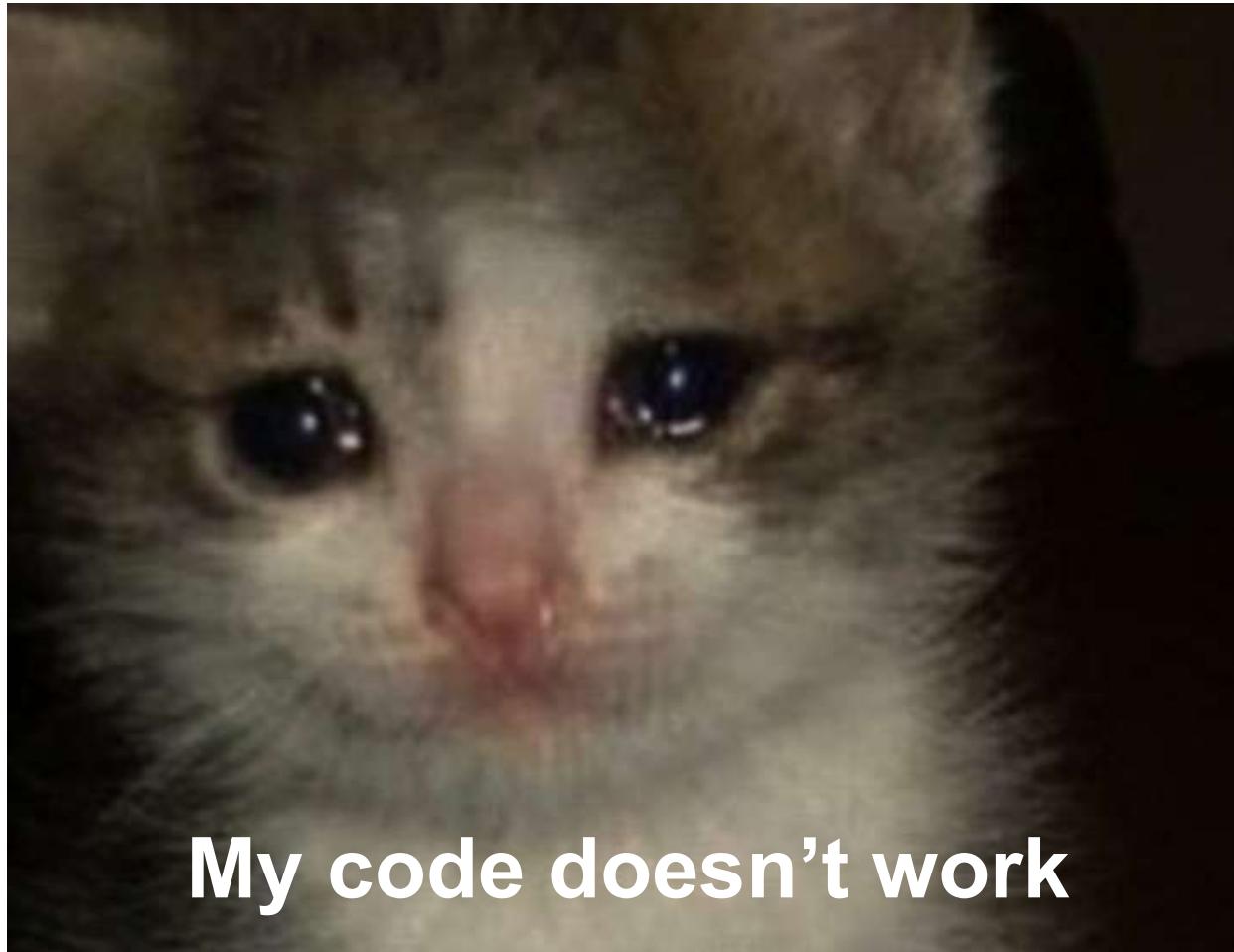
Precipitacion anual



Elevation-Rainfall relationship



# *3. Análisis exploratorio de datos*



**My code doesn't work**

```
read.table(file, header = FALSE, sep = "", quote = "\"\"", dec = ".", row.names, col.names, as.is = FALSE, na.strings = "NA", colClasses = NA, nrows = -1, skip = 0, check.names = TRUE, fill = !blank.lines.skip, strip.white = FALSE, blank.lines.skip = TRUE, comment.char = "#")
```

```
read.csv(file, header = TRUE, sep = ",", quote = "\"\"", dec = ".", fill = TRUE, ...)
```

<b>file</b>	the name of the file (within "" or a variable of mode character), possibly with its path (the symbol \ is not allowed and must be replaced by /, even under Windows), or a remote access to a file of type URL ( <a href="http://...">http://...</a> )
<b>header</b>	a logical (FALSE or TRUE) indicating if the file contains the names of the variables on its first line
<b>sep</b>	the field separator used in the file, for instance <b>sep="\t"</b> if it is a tabulation
<b>quote</b>	the characters used to cite the variables of mode character
<b>dec</b>	the character used for the decimal point
<b>row.names</b>	a vector with the names of the lines which can be either a vector of mode character, or the number (or the name) of a variable of the file (by default: 1, 2, 3, ...)
<b>col.names</b>	a vector with the names of the variables (by default: V1, V2, V3, ...)
<b>as.is</b>	controls the conversion of character variables as factors (if FALSE) or keeps them as characters (TRUE); <b>as.is</b> can be a logical, numeric or character vector specifying the variables to be kept as character
<b>na.strings</b>	the value given to missing data (converted as NA)
<b>colClasses</b>	a vector of mode character giving the classes to attribute to the columns
<b>nrows</b>	the maximum number of lines to read (negative values are ignored)
<b>skip</b>	the number of lines to be skipped before reading the data
<b>check.names</b>	if TRUE, checks that the variable names are valid for R
<b>fill</b>	if TRUE and all lines do not have the same number of variables, "blanks" are added
<b>strip.white</b>	(conditional to <b>sep</b> ) if TRUE, deletes extra spaces before and after the character variables
<b>blank.lines.skip</b>	if TRUE, ignores "blank" lines
<b>comment.char</b>	a character defining comments in the data file, the rest of the line after this character is ignored (to disable this argument, use <b>comment.char = ""</b> )

## Leyendo archivos, bases de datos

## *Configurando Series de tiempo*

```
ts(data = NA, start = 1, end = numeric(0), frequency = 1, deltat = 1, ts.eps =  
getOption("ts.eps"), class, names)
```

data	a vector or a matrix
start	the time of the first observation, either a number, or a vector of two integers (see the examples below)
end	the time of the last observation specified in the same way than start
frequency	the number of observations per time unit
deltat	the fraction of the sampling period between successive observations (ex. 1/12 for monthly data); only one of frequency or deltat must be given
ts.eps	tolerance for the comparison of series. The frequencies are considered equal if their difference is less than ts.eps
class	class to give to the object; the default is "ts" for a single series, and c("mts", "ts") for a multivariate series
names	a vector of mode character with the names of the individual series in the case of a multivariate series; by default the names of the columns of data, or Series 1, Series 2, ...

## Zoo y xts

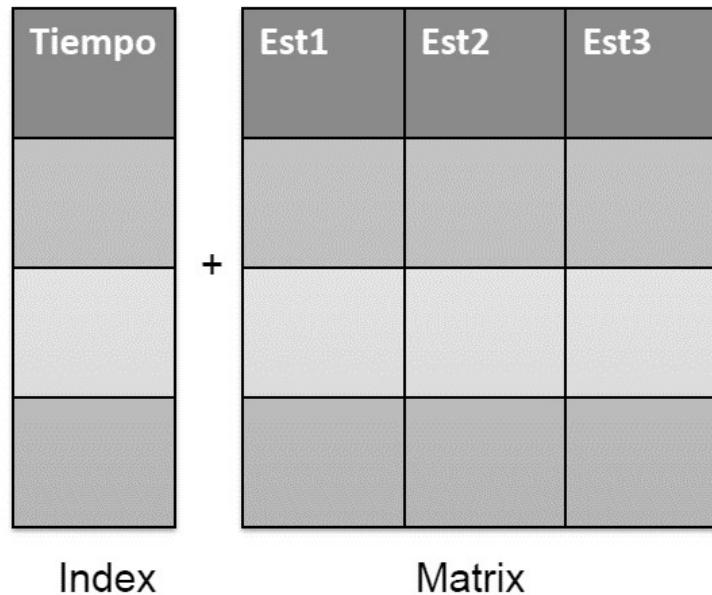
Z's Ordered Observations

```
zoo(x = NULL, order.by = index(x), frequency = NULL, calendar =  
getOption("zoo.calendar", TRUE))
```

```
# S3 method for zoo
```

```
print(x, style = , quote = FALSE, ...)
```

```
xts(x = NULL, order.by = index(x), frequency = NULL, unique = TRUE, tzone =  
Sys.getenv("TZ"), ...)  
is.xts(x)
```



## B. Explorando datos de precipitación diaria de varias estaciones almacenados en una matriz

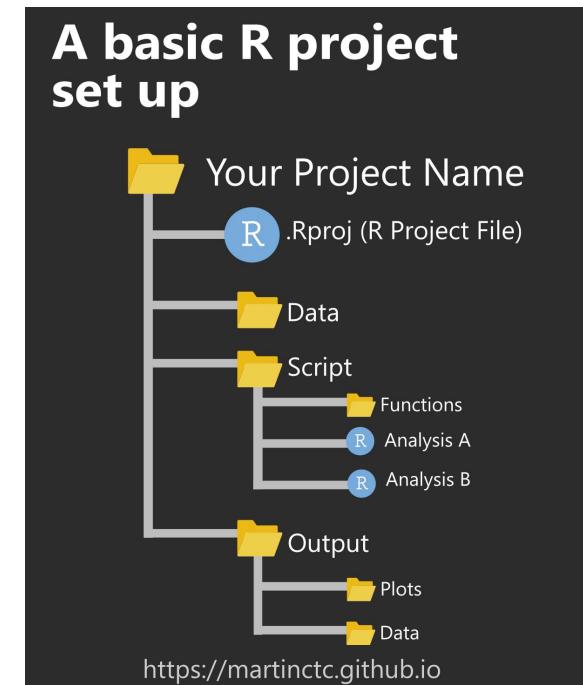
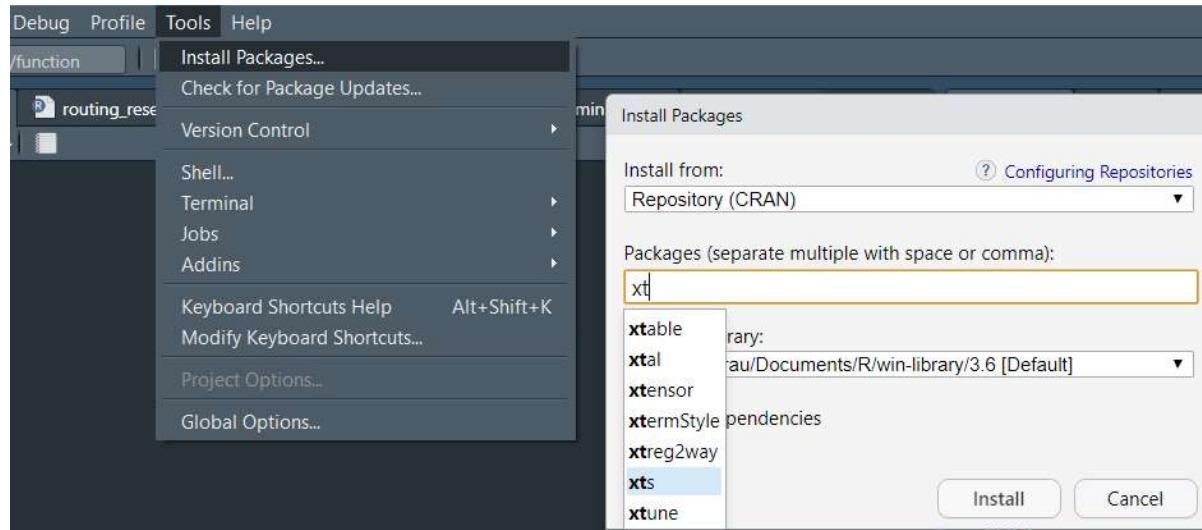
*Uso de las librerías: zoo, xts, lattice, ggplot2, visdat*

-Instalar librerías TOOLS - INSTALL PACKAGES: **zoo**, **xts**, **lattice**, **ggplot2**, **visdat**

-Descargar archivo "**p\_diarias.csv**"

[https://raw.githubusercontent.com/hydrocodes/hidRologia\\_corto/master/p\\_diarias.csv](https://raw.githubusercontent.com/hydrocodes/hidRologia_corto/master/p_diarias.csv)

-Revisión de archivo csv (separados por comas con formato de fecha %Y-%m-%d)



## Ejercicio C

En la **consola de scripts**, crear un código que realice lo siguiente:

- 1) Leer el archivo de precipitaciones diarias "p\_diarias.csv" almacenados en formato matriz con 4 estaciones (identificadores 101, 102, 103, 104) desde el 1Ene2005 al 31Dic2014
- 2) Plotear las series de tiempo para cada estación usando el Rbase, lattice y ggplot.
- 3) Plotear las series de tiempo en conjunto con la misma escala vertical
- 4) Plotear un boxplot de las estaciones en conjunto
- 5) Plotear 3 histogramas correspondientes a 3 estaciones
- 6) Visualizar un diagrama de calor de datos faltantes de todas las estaciones

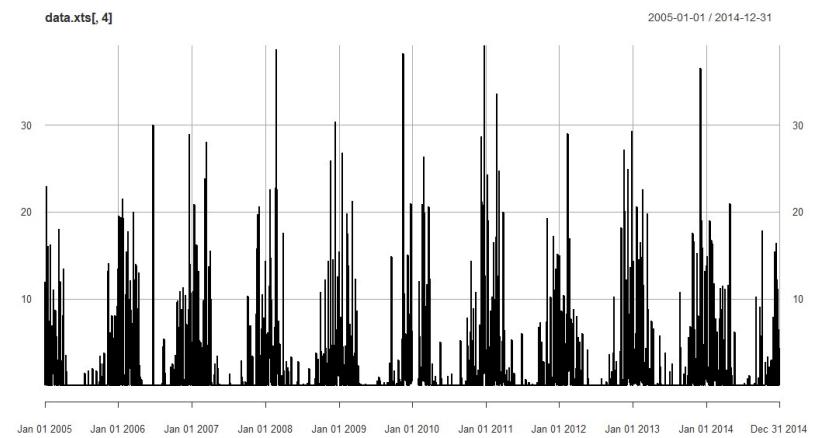
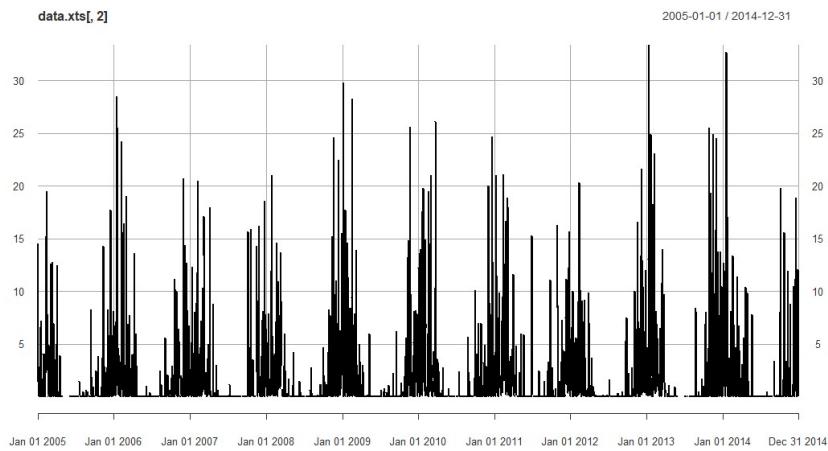
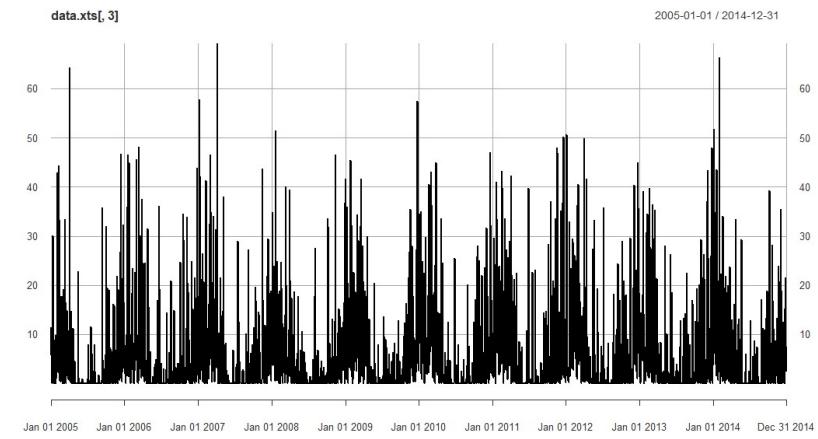
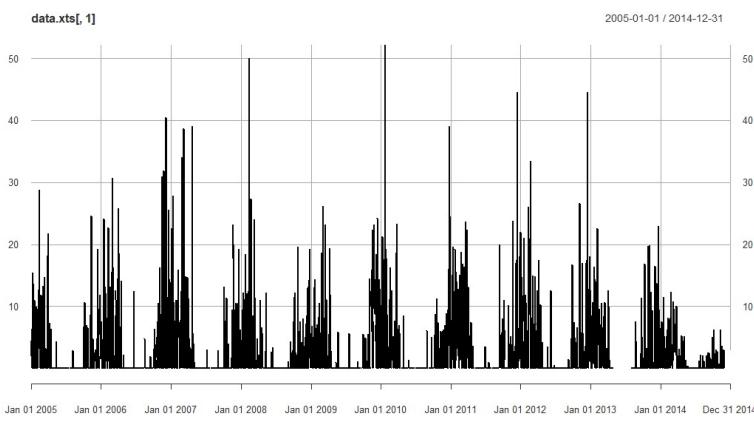
# Respuestas C

```
library(zoo)
library(xts)
library(lattice)
library(ggplot2)
pdiaria <- read.csv(file.choose(), header=TRUE, check.names = F, stringsAsFactors = F)
str(pdiaria) #para ver la estructura del objeto
idx <- as.Date(pdiaria[,1]) #formato fecha a la 1era columna
data.matrix <- pdiaria[,-1] #formato matriz al resto de columnas a excepcion de la 1era columna
data.xts <- xts(data.matrix, order.by = idx ) #crear un objeto xts (eXtended time series)
str(data.xts)
plot(data.xts)
plot(data.matrix[,1], type="l")
plot(data.xts[,2])

# convertir a un objeto zoo
data.zoo <- as.zoo(data.xts)
str(data.zoo)
plot(data.zoo, main = "Series de tiempo de precipitación")
summary(data.zoo)
max(data.zoo, na.rm = T)
plot(data.zoo, main = "Series de tiempo de precipitación", ylim = c(0,80))

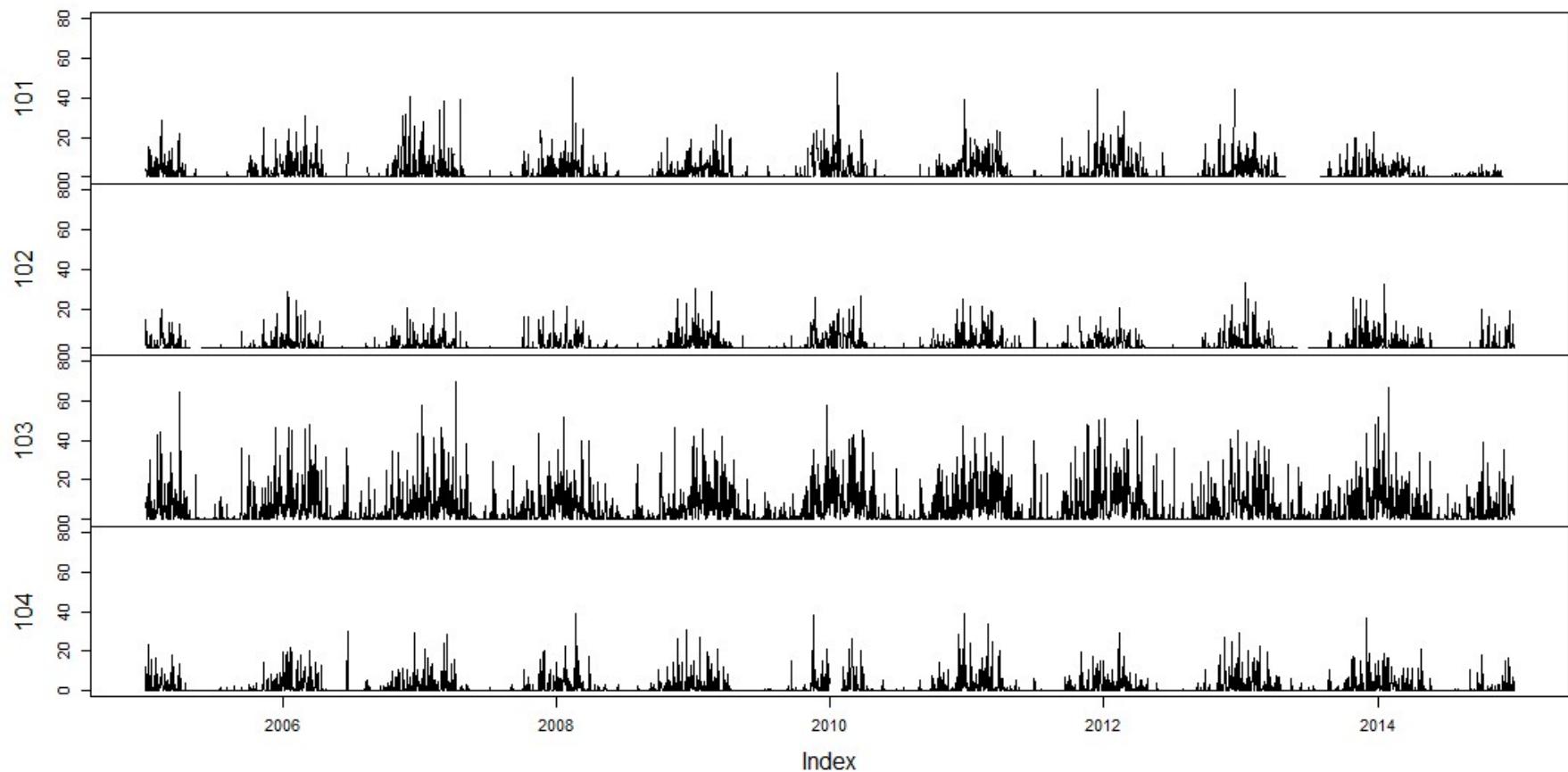
xyplot(data.xts, xlab = "Fecha", ylab = "Precipitación [mm/dia]", ylim=c(0,100))
ggplot(data=pdiaria, aes(x=as.Date(fecha), y=pdiaria$'101')) + geom_line() + theme_bw() + xlab('Fecha') +
ylab('Precipitación [mm/dia]')
autoplot(data.xts[,1:2]) + theme_bw() + xlab('Fecha') + ylab('Precipitación [mm/dia]')
boxplot(coredata(data.xts))
hist(coredata(data.xts[,1]), freq = T) # cantidad de datos por clase
histogram(coredata(data.xts[,1])) # porcentaje de datos por clase

# Heatmap de valores faltantes
library(visdat)
vis_miss(pdiaria, sort_miss = TRUE)
```

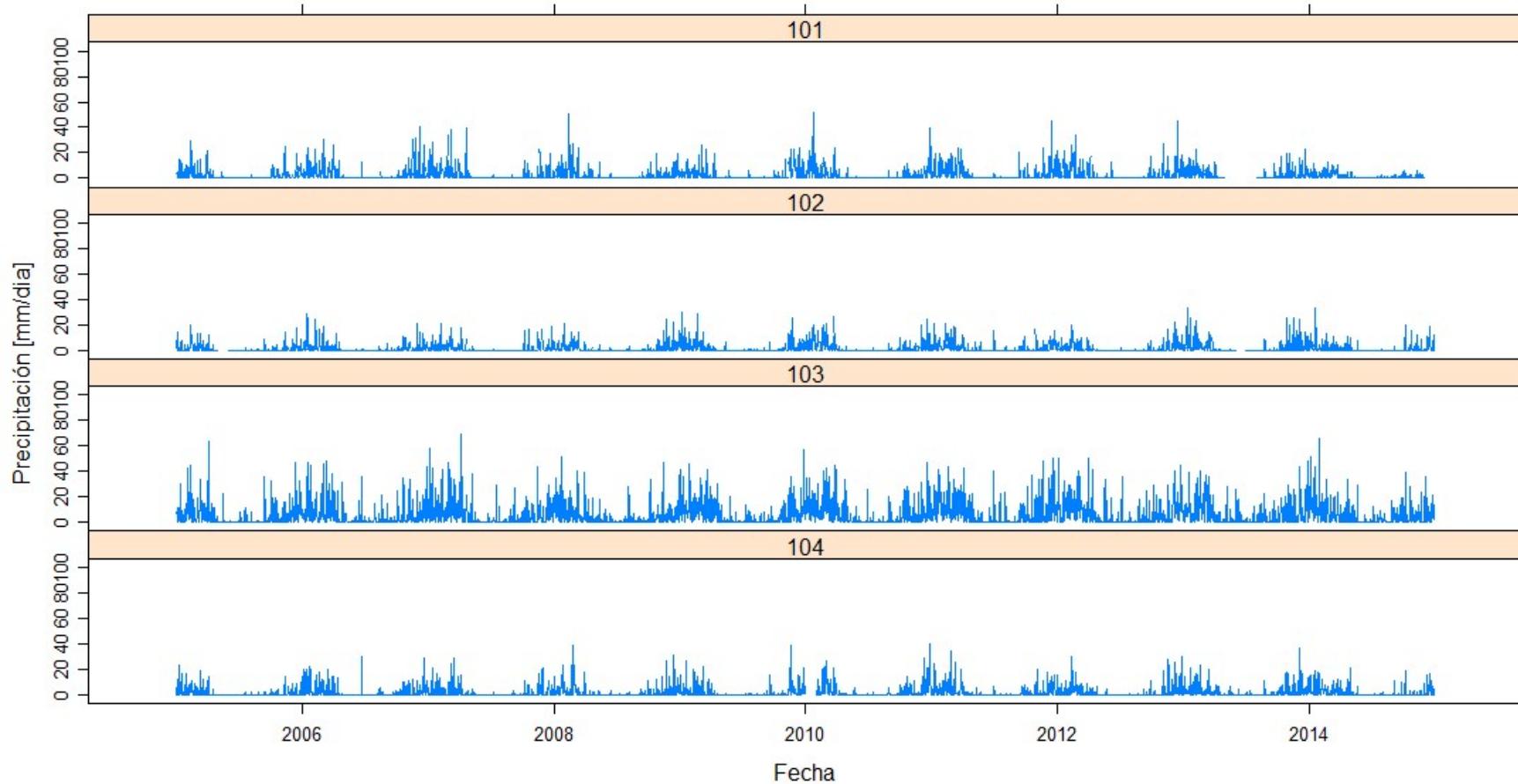


## *Ploteo simple*

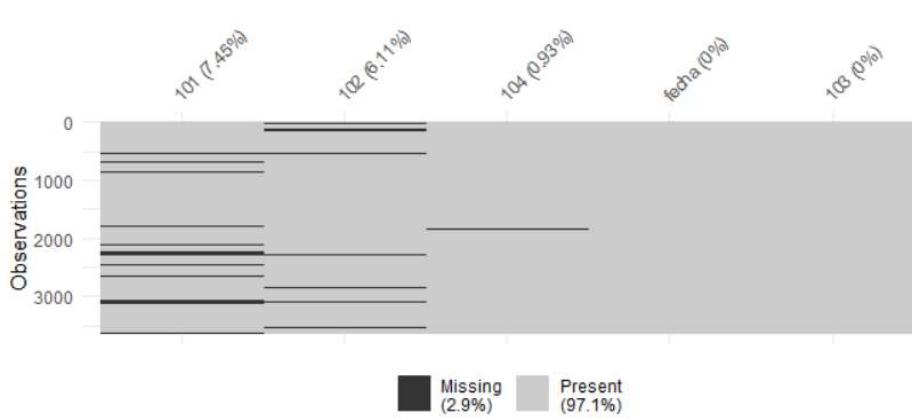
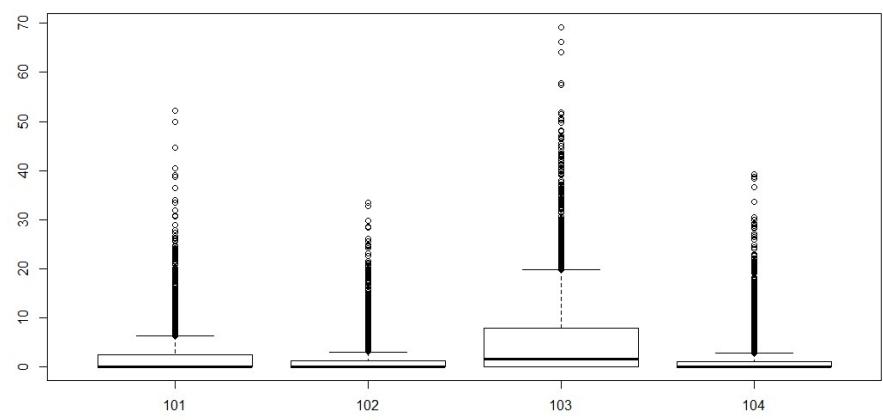
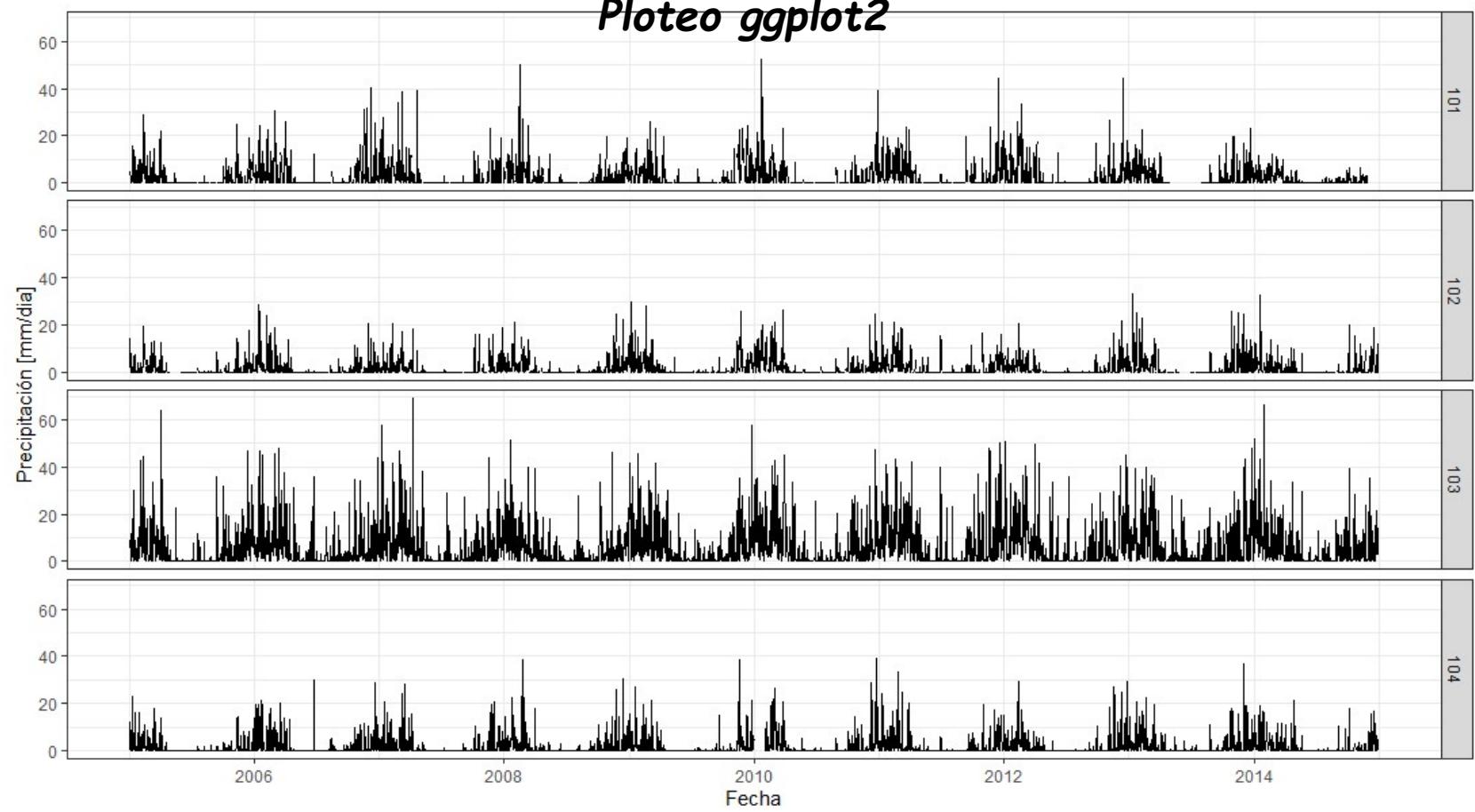
Series de tiempo de precipitación

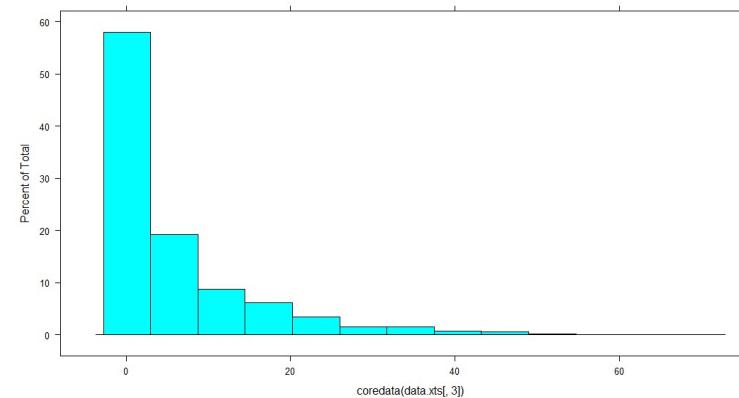
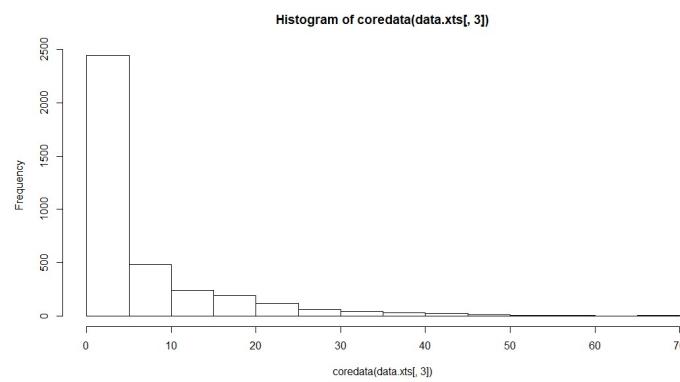
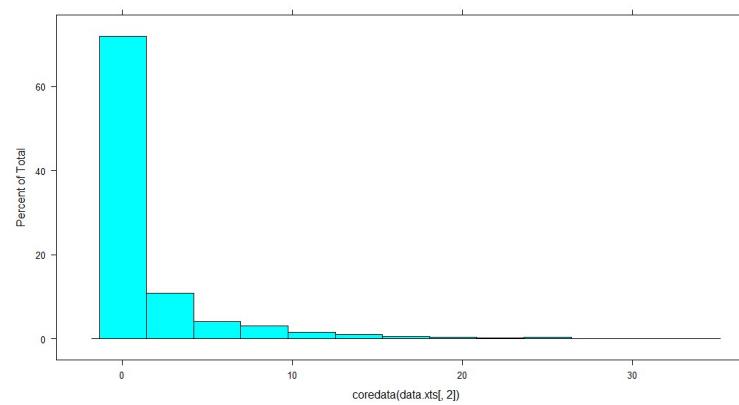
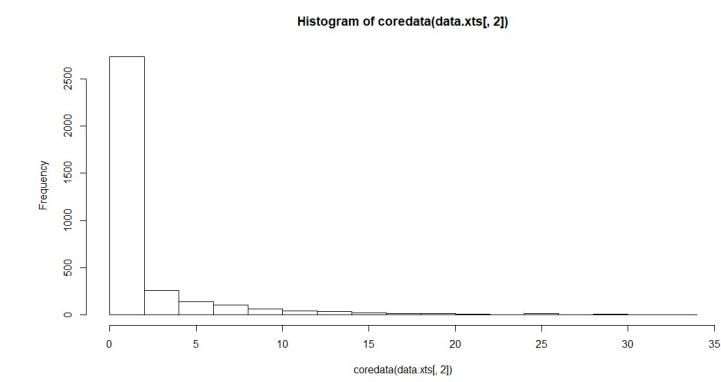
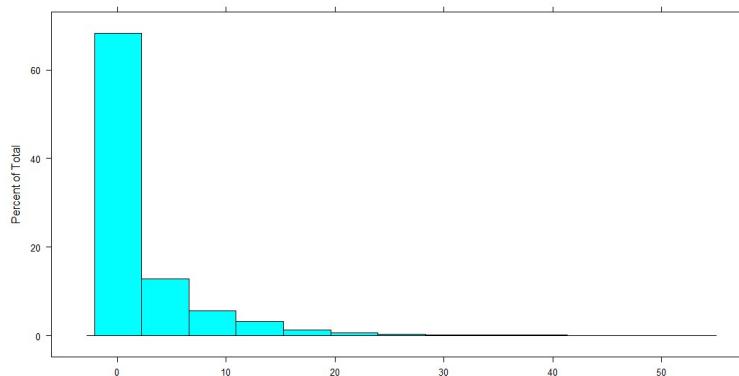
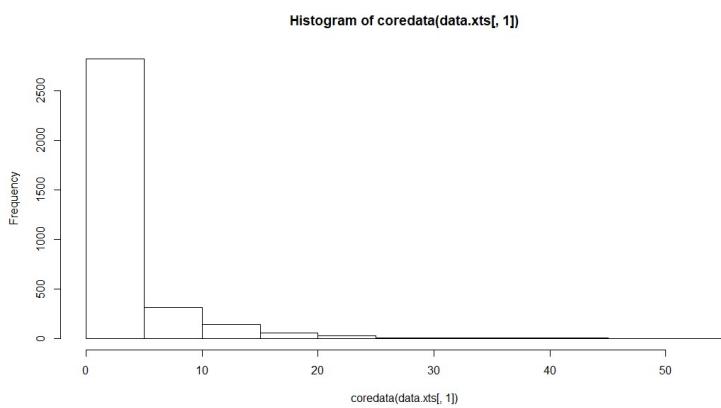


## *Ploteo lattice*



# Ploteo ggplot2



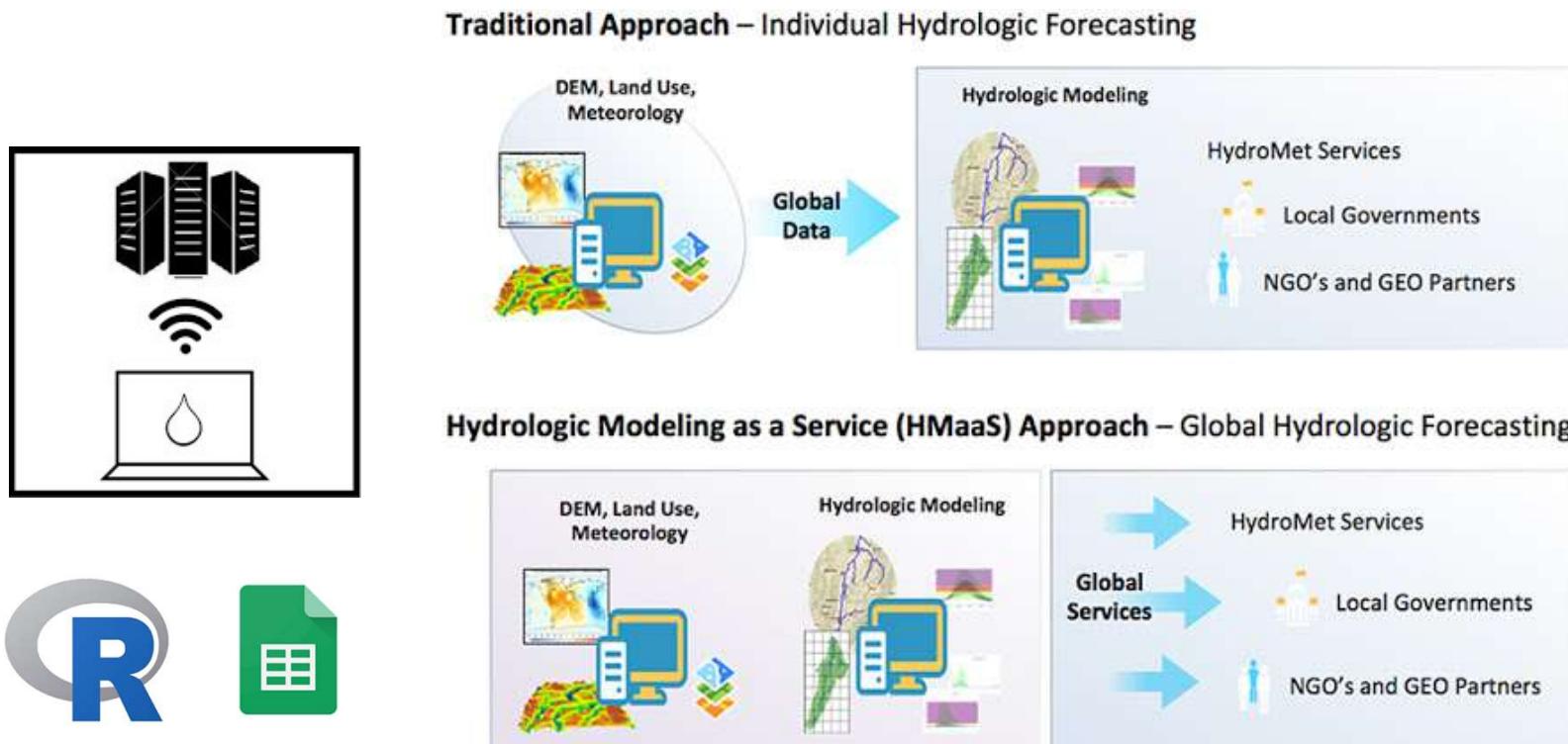


# *4. Introducción a la hidrología en la nube*



# El *cloud computing* o la *computación desde la nube*.

Si bien, la hidroinformática es la especialidad que se encarga de los sistemas de información y comunicaciones aplicados a la gestión y manejo del agua, esta nueva forma de prestación de servicios en tecnología, viene complementadola en el manejo del bigdata en hidrología. Cuestión de imaginar al **Software**, a la **Plataforma** y a la **Infraestructura** como un servicio (SaaS, Paas, IaaS) y no como algo propio. De esta forma es posible incluso administrar grandes cantidades de información desde un celular con una buena conexión a internet.



# Construyendo una Curva de duración de caudales

- Ordenar los caudales de **mayor a menor**,
- Asignar rangos consecutivos desde 1 para el caudal superior, hasta "m" para el caudal inferior.
- Calcular la probabilidad de excedencia o de exceso.

Calcular el **caudal** en función de su % de exceso o "frecuencia de su ocurrencia" o nivel de persistencia.

## La Probabilidad de Excedencia %

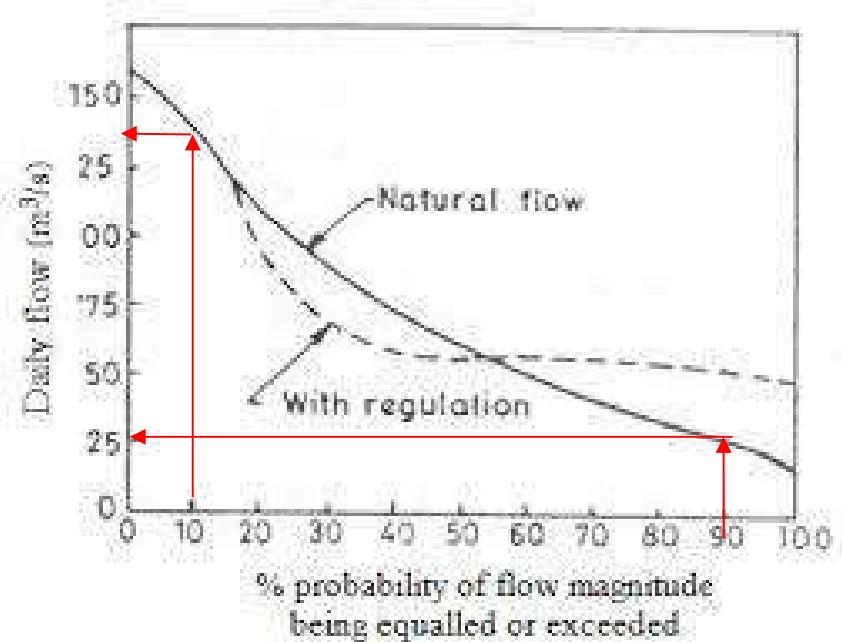
$$P = 100 * \frac{m}{n + 1}$$

m: rango de los caudales desde el más alto al más bajo.

n: número total de datos.



Represa Antacoto - Marca III



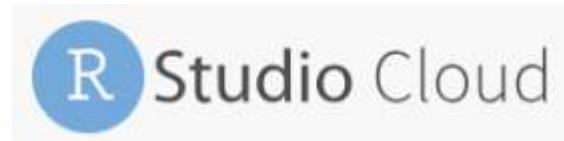
Q10%: Caudal de aguas altas  
Q90%: Caudal de aguas bajas

## D. Trabajando desde GitHub y RstudioCloud

- <https://rstudio.cloud/>
- Instalar la librería Rcurl, hydroTSM
- Leer archivo desde GitHub



**GitHub**

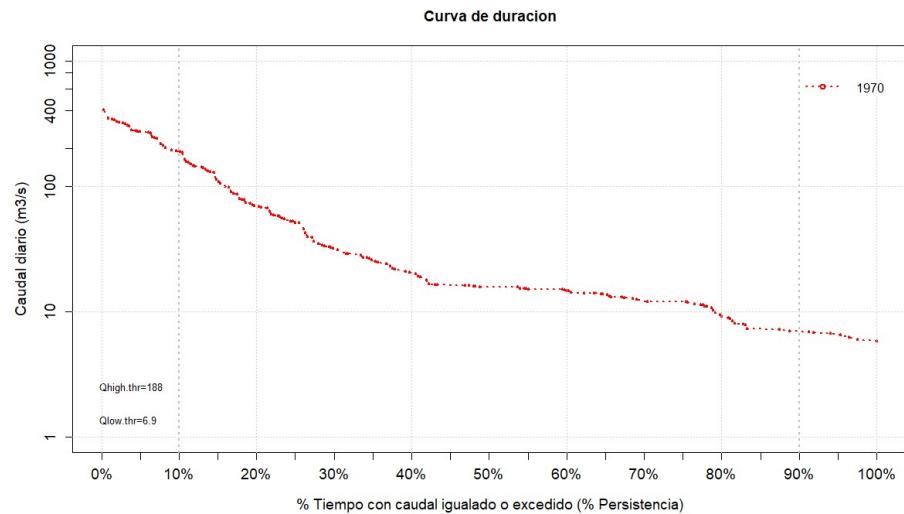


### Ejercicio D: Generando una curva de duración

- 1) Crearse una cuenta en <https://github.com/>
- 2) Cerrar el R y Rstudio, ahora se trabajará únicamente en un entorno cloud. Abrir el RstudioCloud
- 3) Leer el archivo de caudales almacenado en github: caudales.csv  
[https://raw.githubusercontent.com/hidRologia\\_corto/master/caudales.csv](https://raw.githubusercontent.com/hydrocodes/hidRologia_corto/master/caudales.csv)
- 4) Generar la curva de duración y estimar el caudal al 10% y 90% de persistencia

# Respuestas D

```
#### Curva de duracion de caudales con datos online
#### https://github.com/hydrocodes
rm(list = ls())
dev.off()
library(hydroTSM)
library(RCurl)
p<-getURL("https://raw.githubusercontent.com/hydrocodes/hidRologia_corto/master/caudales.csv")
datos<-read.csv(text = p)
qdiario <- datos$Qd_m3s
fdc(qdiario,
  IQ.thr=0.9, hQ.thr=0.1,
  plot=TRUE, log="y",
  main= "Curva de duracion",
  xlab="% Tiempo con caudal igualado o excedido (% Persistencia)",
  ylab="Caudal diario (m3/s)",
  ylim=c(1,1000),
  yat=c(0.1, 1, 10, 100, 1000),
  col="red", pch=21, lwd=2,
  lty=3, cex=0.4,
  cex.axis=1.2, cex.lab=1.2, leg.txt="1970",
  leg.cex=1, leg.pos="topright",
  verbose=TRUE,
  thr.shw=TRUE,
  new=T)
grid(nx = NULL, ny = NULL,
      col = "lightgray", lty =21, lwd=0.5,
      equilogs = TRUE)
```



# Referencias

- Chow V, Maidment D, Mays L. 1994. Hidrologia Aplicada. McGraw-Hill.
- Ihaka R. & Gentleman R. 1996. R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5: 299-314.
- Naghettini M. 2017. Fundamentals of Statistical Hydrology. Springer.
- Paradis E. 2002. R for beginners. Institut de Sciences de l'évolution. Université de Montpellier. France
- Rau P, Bourrel L, Labat D, et al. 2017. Regionalization of rainfall over the Peruvian Pacific slope and coast. *International Journal of Climatology* 37(1):143-158.
- RStudio Team (2015). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA
- Slater L, Thirel G, Harrigan S et al. 2019. Using R in hydrology: a review of recent developments and future directions. *Hydrol. Earth Syst. Sci.*, 23, 2939-2963
- Souffront Alcantara MA, Nelson EJ, Shakya K, Edwards C, Roberts W, Krewson C, Ames DP, Jones NL and Gutierrez A (2019) Hydrologic Modeling as a Service (HMaaS): A New Approach to Address Hydroinformatic Challenges in Developing Countries. *Front. Environ. Sci.* 7:158. doi: 10.3389/fenvs.2019.00158