Taller corto de verano

Short summer workshop

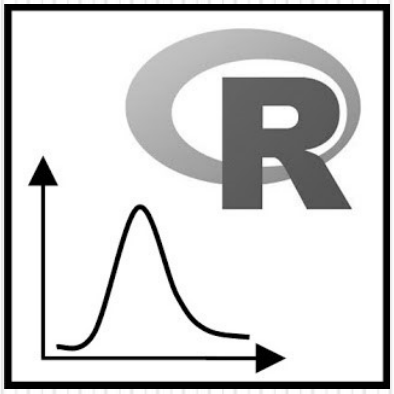
Lima, 31 enero 2020

Introducción al análisis de series hidrológicas en R

Introduction to hydrological series analysis in R

Pedro Rau, PhD

Hidrólogo



- ✓ **Instructores:**

  Pedro Rau

  - PhD en Hidrología – Universidad de Toulouse - Francia
  - Profesor full-time e investigador Universidad de Ingeniería y Tecnología UTEC – Dpto. Ambiental. Centro de Investigación y Tecnología del Agua CITA.
  - Profesor visitante UNI – Postgrado FIC.
  - Hidrólogo investigador en redes internacionales: FR, EC, US, UK
  - Consultor en Hidrología y Recursos Hídricos

  Erick Claros

  - Bach. Ing. Civil – Universidad Nacional de Ingeniería UNI
  - Consultor en Hidráulica y Recursos Hídricos

- ✓ **E.mail:** praul@uni.pe

- ✓ **Sitio web:** http://pedrorau.blogspot.com

- ✓ **Repositorio del curso:** https://github.com/hydrocodes/Intro_R

- ✓ **Requisitos:** Conocimientos básicos en Hidrología y Estadística

Objetivos del curso

- Brindar las bases teóricas del estudio de las series de tiempo hidrológicas.

- Punto de inicio para los interesados en el sistema R y su entorno estadístico.

- Desarrollar capacidades para el empleo correcto del software libre en la hidrología.

Temáticas

1. Introducción a las series de tiempo hidrológicas
2. Entorno R y Rstudio
3. Análisis exploratorio de datos

Evaluación

Clases teóricas + prácticas
- Asistencia y desarrollo de ejercicios
  5 hr presenciales + 1hr virtual

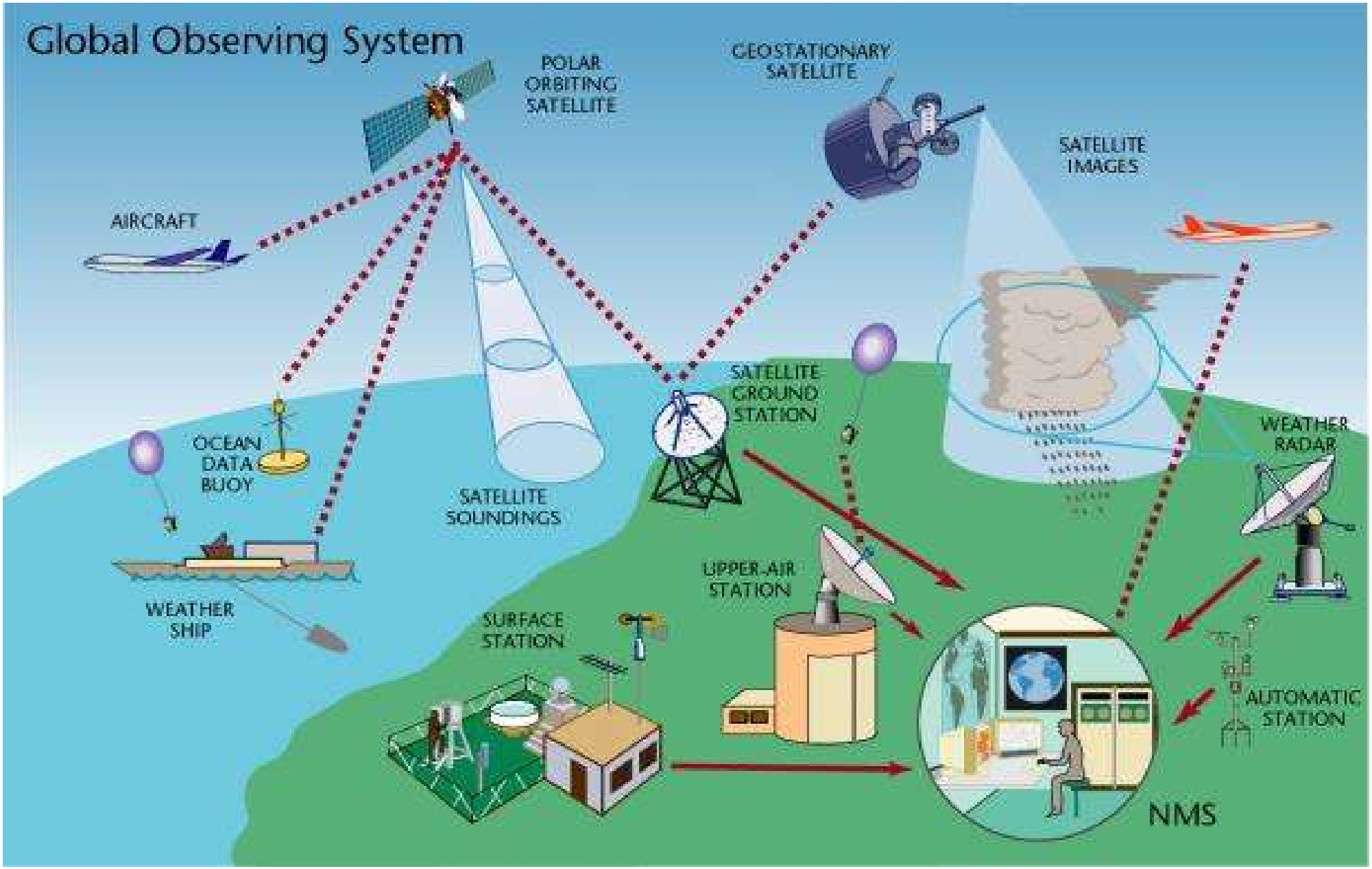
Metodología « Hands-on »

- Lecturas y papers base
- Planteamiento de ejercicios
- Creación/tipeo/interpretación de códigos *.R con semi comentarios
- Resolución de bugs
- Finalización de códigos *.R con comentarios completos al detalle (evaluado)

Total: 6hr lectivas

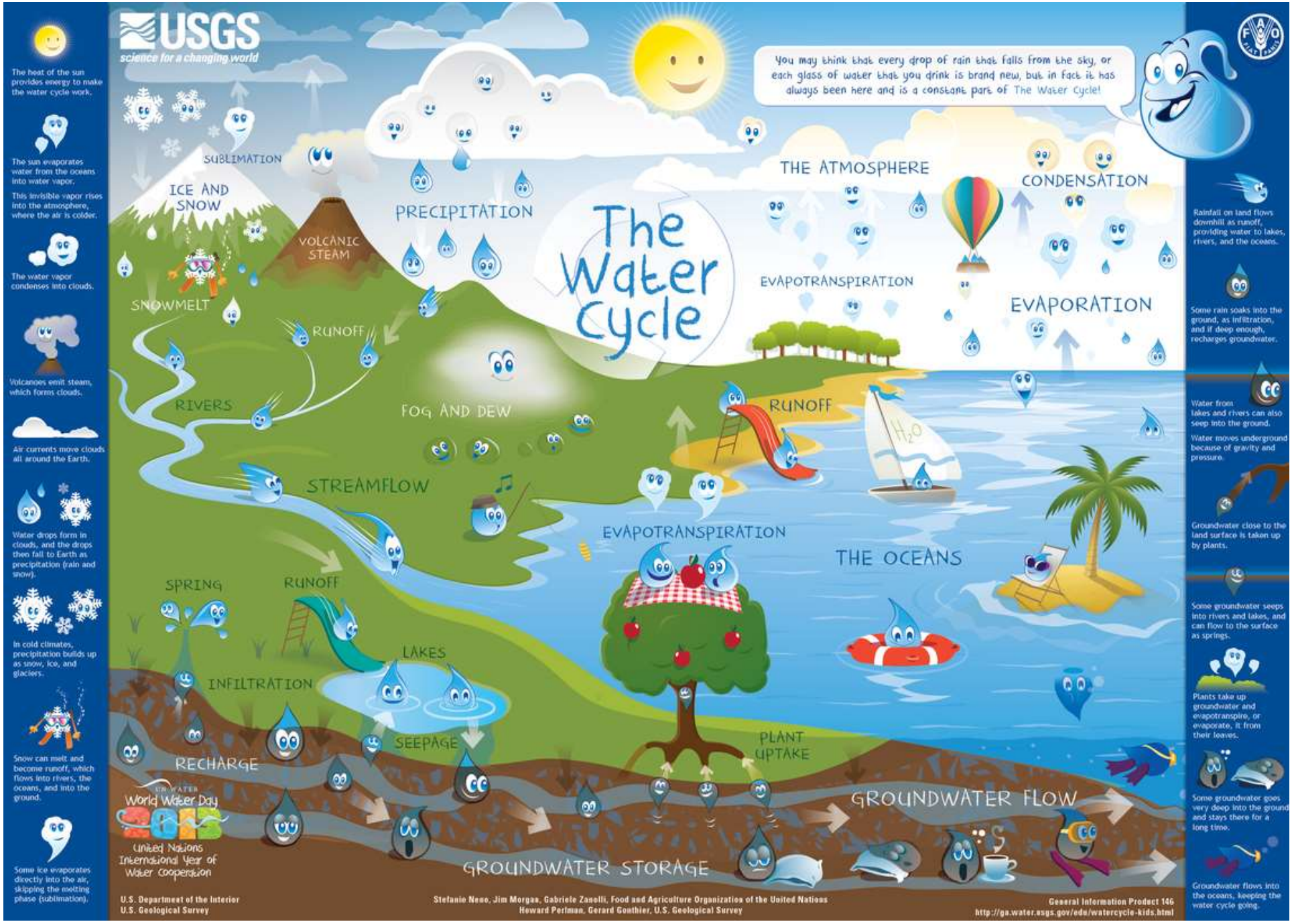
1. Introducción a las series de tiempo hidrológicas

1.1 Sistema de observación global



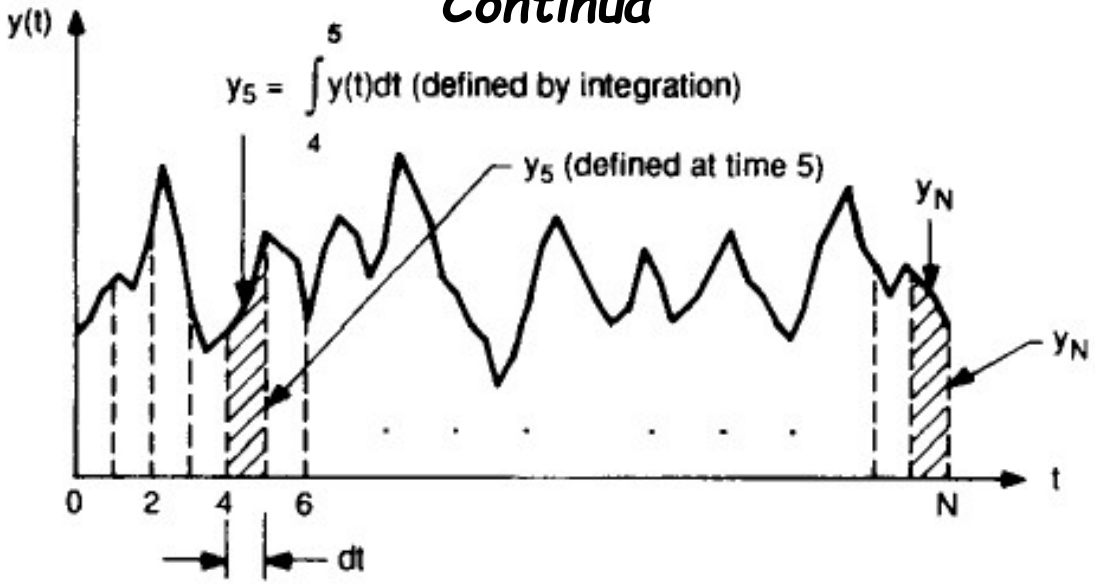
WMO

1.2 El ciclo hidrológico

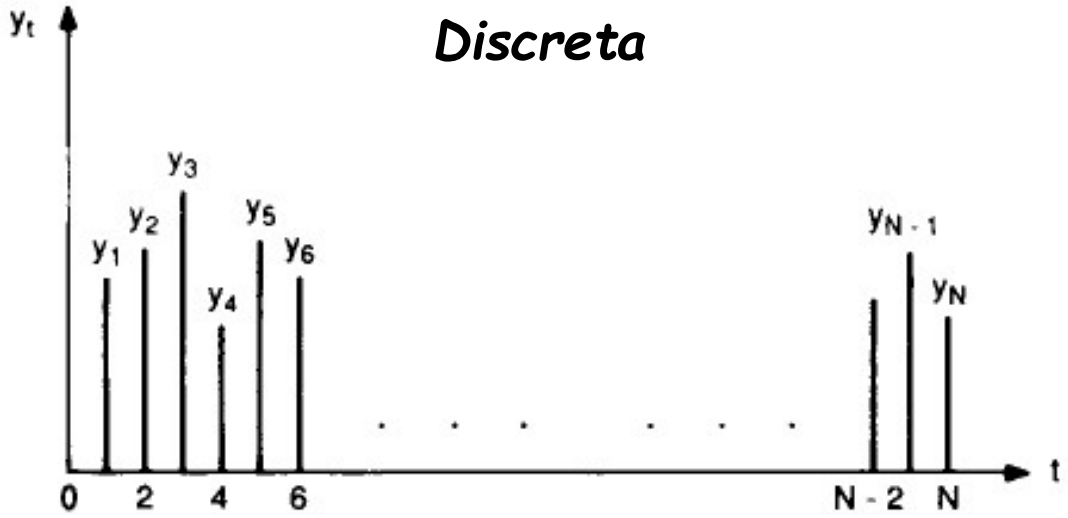


1.3 Tipos o categorías de series de tiempo

Continua



Discreta

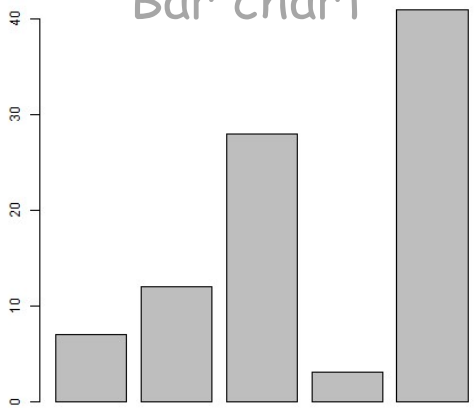


- ✓ Unicas
- ✓ Multiples
- ✓ Independientes, no correlacionadas; autocorrelacionados o dependientes
- ✓ Intermitentes
- ✓ De conteo
- ✓ Regulares o irregulares en espacios de tiempo
- ✓ Estacionarias (sin tendencia, ni salto ni ciclicidad / periodicidad) y no-estacionarias

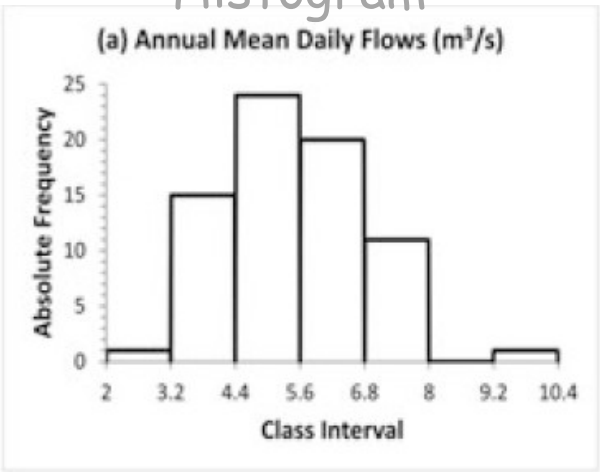
Salas (1996)

1.6 Análisis preliminar de datos hidrológicos
a. Representación gráfica

Diagrama de barras
Bar chart



Histograma
Histogram



Poligonos de frecuencia
Frequency Polygon

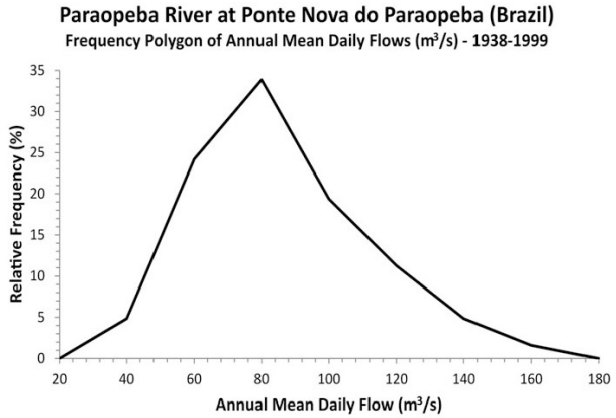
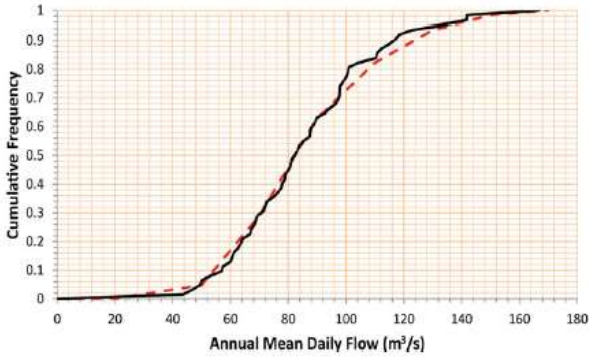
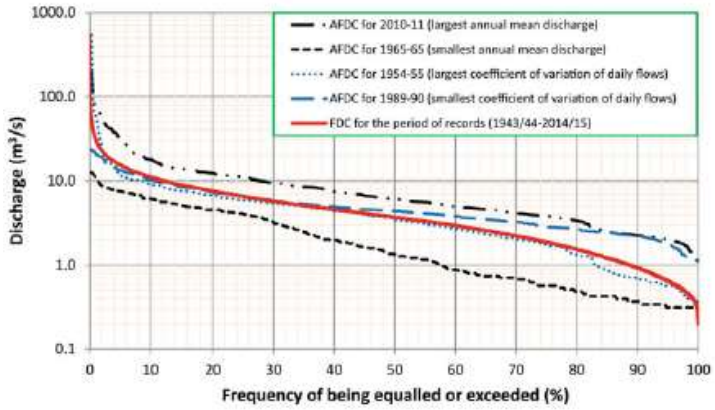


Diagrama de puntos
Dot chart

Frecuencia relativa acumulada
Cumulative Relative Frequency Diagram



Curvas de duración
Duration curves



Naghettini (2017)

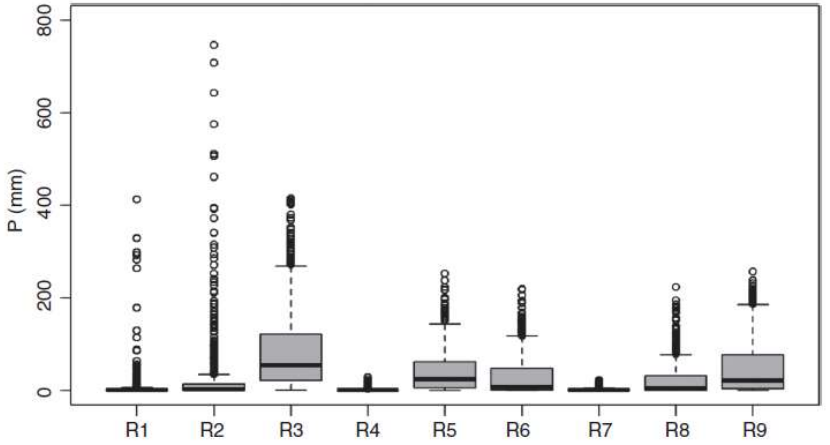
b. Estadísticos descriptivos

Parámetro de la población	Estadística de la muestra
1. *Punto medio*	
Media aritmética	
$\mu = E(X) = \int_{-\infty}^{\infty} x f(x)\,dx$	$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$
Mediana	
x tal que $F(x) = 0.5$	Valor de la información en el 50o. percentil
Media geométrica	
antilog $[E(\log x)]$	$\left(\prod_{i=1}^{n} x_i\right)^{1/n}$
2. *Variabilidad*	
Varianza	
$\sigma^2 = E[(x-\mu)^2]$	$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2$
Desviación estándar	
$\sigma = \{E[(x-\mu)^2]\}^{1/2}$	$s = \left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2\right]^{1/2}$
Coeficiente de variación	
$CV = \frac{\sigma}{\mu}$	$CV = \frac{s}{\bar{x}}$
3. *Simetría*	
Coeficiente de asimetría (oblicuidad)	
$\gamma = \frac{E[(x-\mu)^3]}{\sigma^3}$	$C_s = \frac{n\sum_{i=1}^{n}(x_i-\bar{x})^3}{(n-1)(n-2)s^3}$

Chow (1994)

c. Métodos exploratorios

Diagrama de cajas
Box plot



Rau et al (2017)

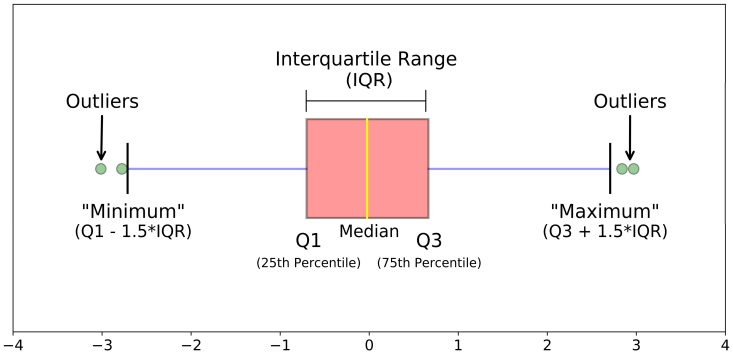
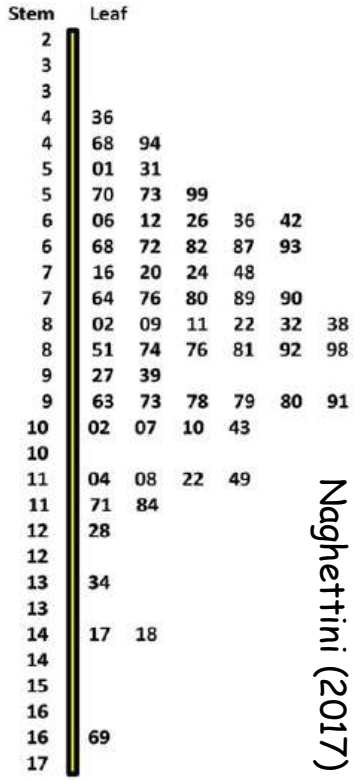


Diagrama de tallo y hojas
Stem-and-leaf Diagram



Naghettini (2017)

d. Asociación de datos

Gráfico de dispersion
Scatter plot

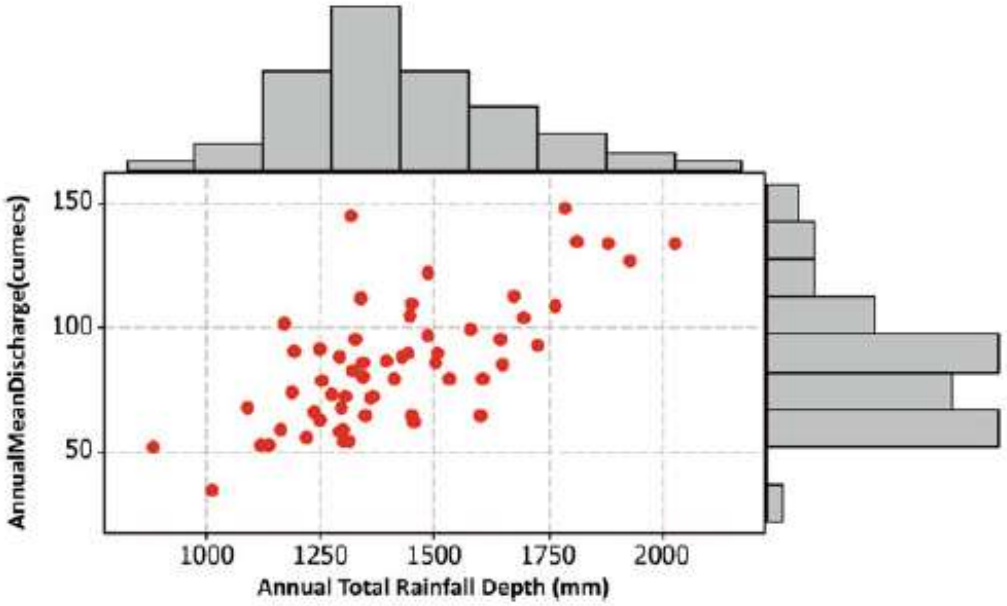
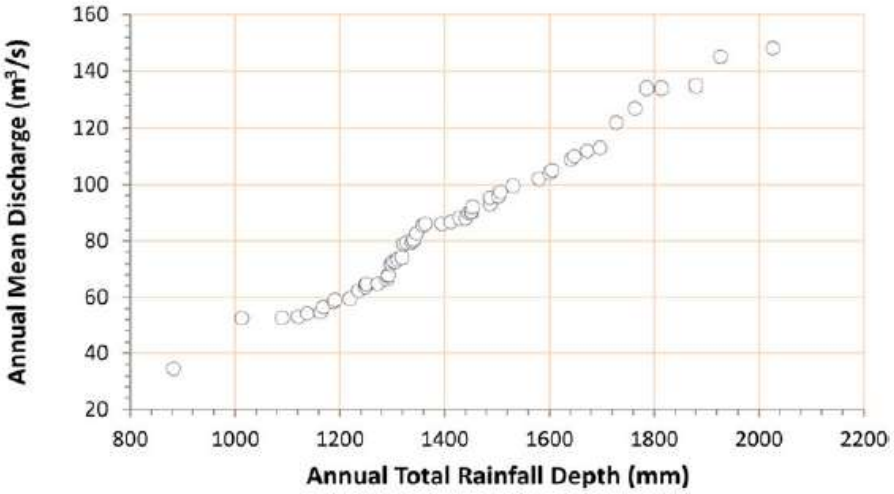
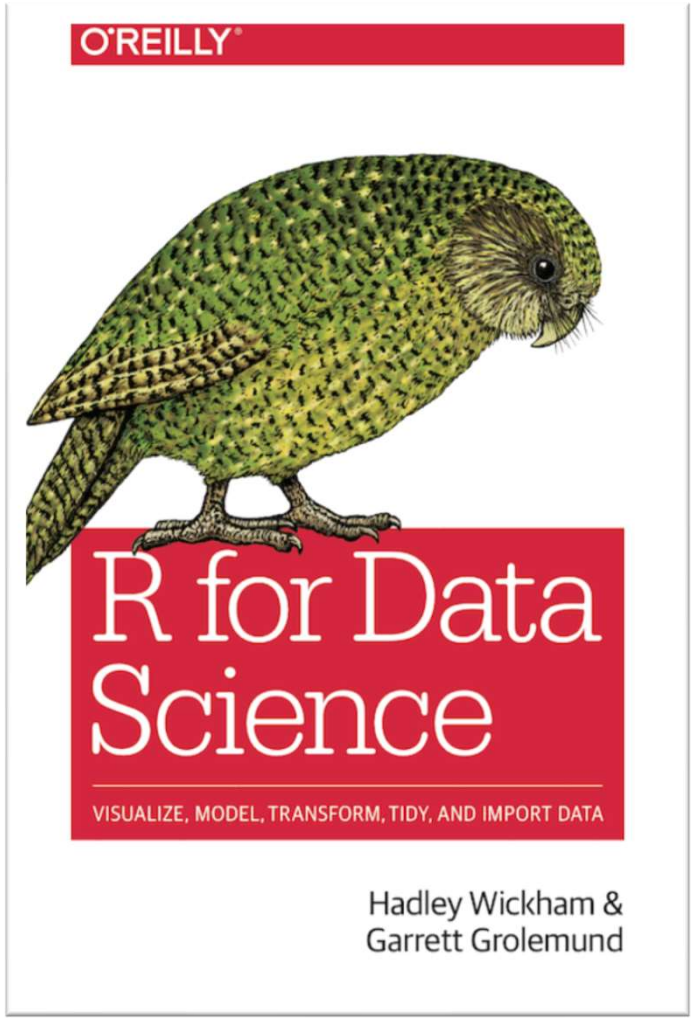


Diagrama Cuantil-Cuantil
Empirical Quantile-Quantile Diagram
Q-Q Plot



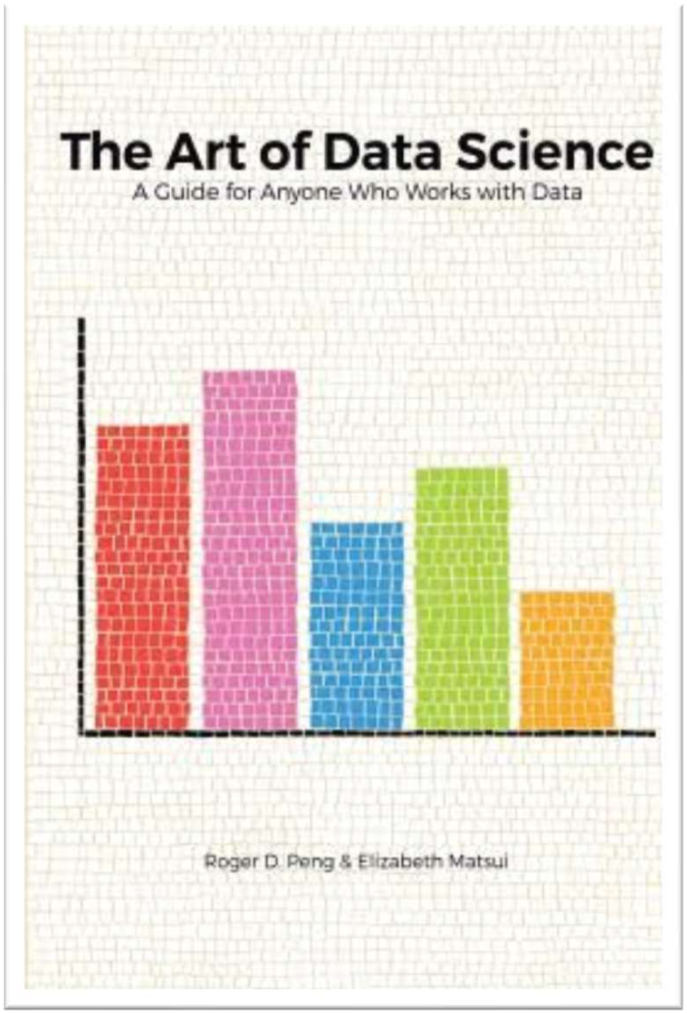
Naghettini (2017)

2. Entorno R y Rstudio



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

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



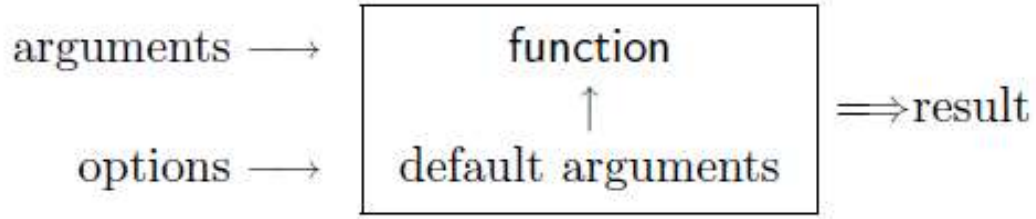
https://bookdown.org/rdpeng/artofdatascience/

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

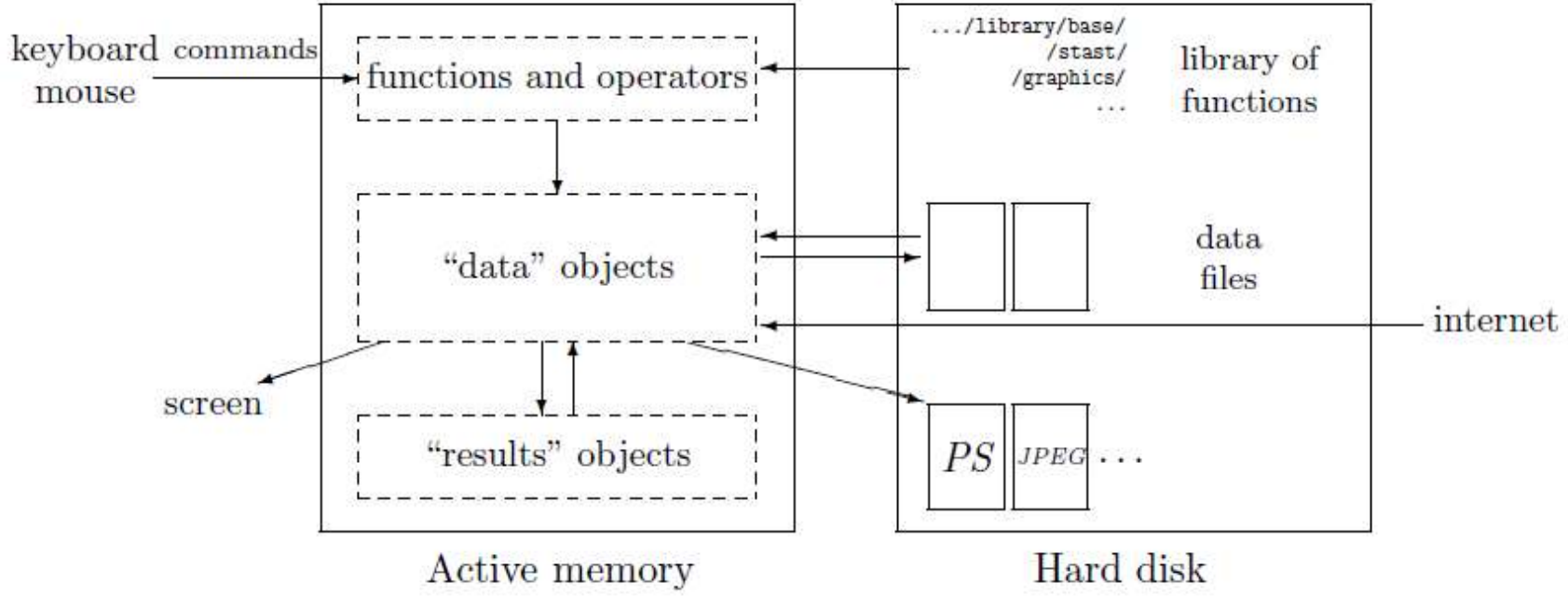
R-3.6.2 for Windows (32/64 bit) – Diciembre 2019

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
- ✓ Facil e intuitivo, estrictamente « no es un lenguaje de programación »

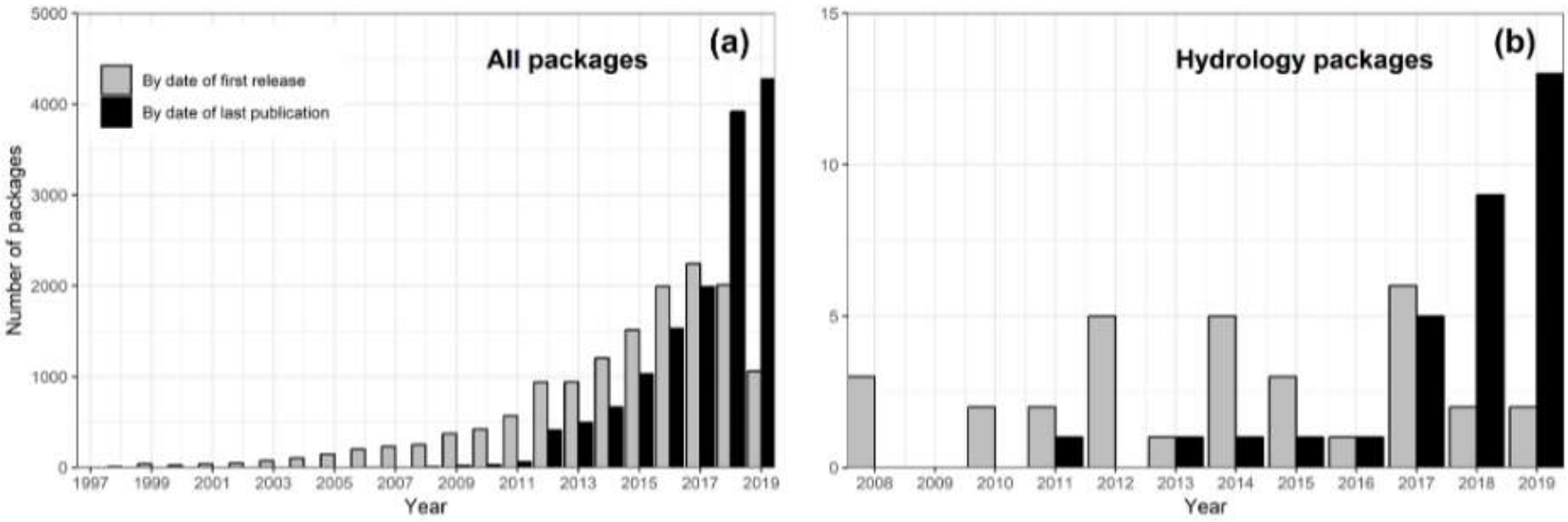


Paradis (2002)

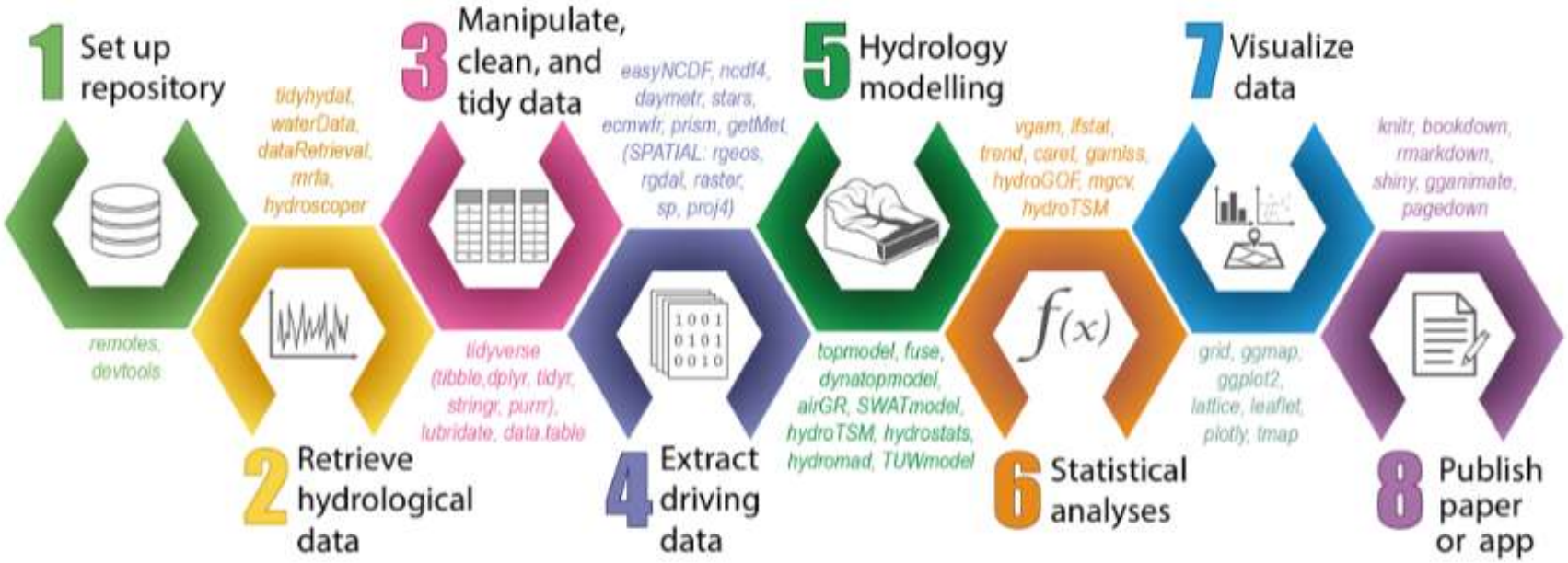


The Comprehensive R Archive Net-work (CRAN)

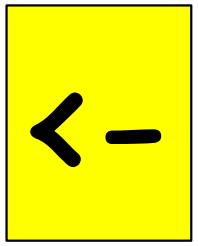
https://cran.r-project.org



Tipico Flujo de trabajo en R en hidrologia

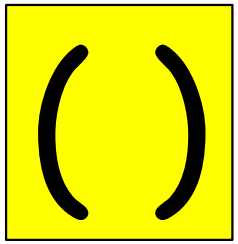


Slater et al (2019)



"Clases" de objetos básicos o "atomic" en R:

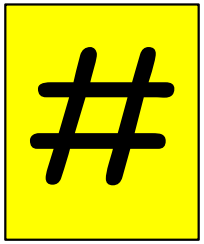
- numeric (real numbers)
- integer
- complex
- logical (True/False)



Tipos de objetos en R para representar a los datos

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

Paradis (2002)



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' (t)	`rt(n, df)`
Fisher–Snedecor (F)	`rf(n, df1, df2)`
Pearson (χ^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	`rnbinom(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				

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

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 1.2.5033

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

Entorno de desarrollo integrado para el lenguaje de programación R

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: 20, 21.2, 49, 10.5, 15.1, 19, 31 y 20.2 en el objeto rain.
9) Visualizar rain
10) Obtener el 5to elemento de rain
11) Almacenar los siguientes valores anuales de índice SOI El Niño: 0.2, 0.4, 1.5, -0.1, 0.15, 0.3, 0.9 y 1.1 en el objeto soi.
12) Visualizar soi.

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(20, 21.2, 49, 10.5, 15.1, 19, 31, 20.2) #Uso de función c(combine)
asignar datos al objeto rain
> rain #Escribimos el objeto "rain"
[1] 20.0 21.2 49.0 10.5 15.1 19.0 31.0 20.2
> rain[5] #Obtenemos el 5to elemento del objeto rain
[1] 15.1
> soi<-c(0.2, 0.4, 1.5, -0.1, 0.15, 0.3, 0.9, 1.1) # Ejemplo de valores del índice
SOI para El Niño
> soi #Escribimos el objeto "soi"
[1]  0.20  0.40  1.50 -0.10  0.15  0.30  0.90  1.10
```

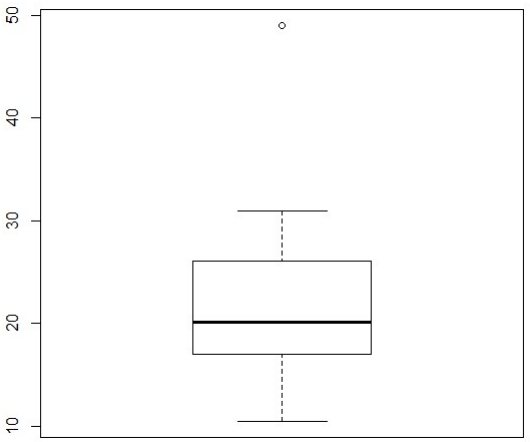
Ejercicio A.2

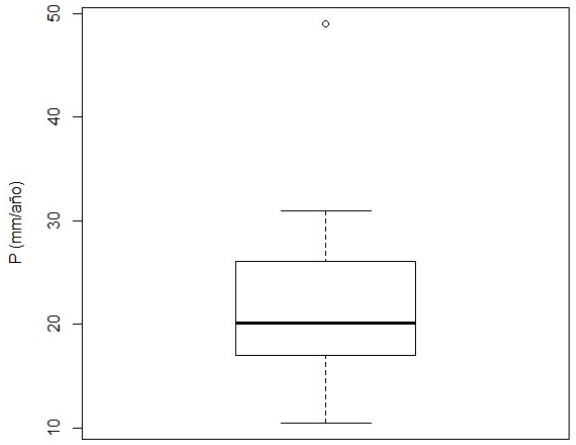
En la consola de resultados, efectuar las siguientes operaciones:

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
```





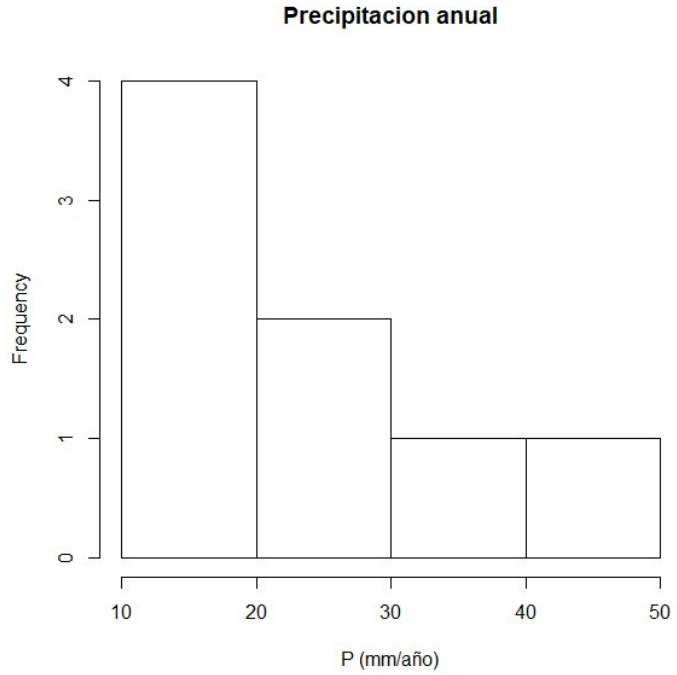
Ejercicio A.3

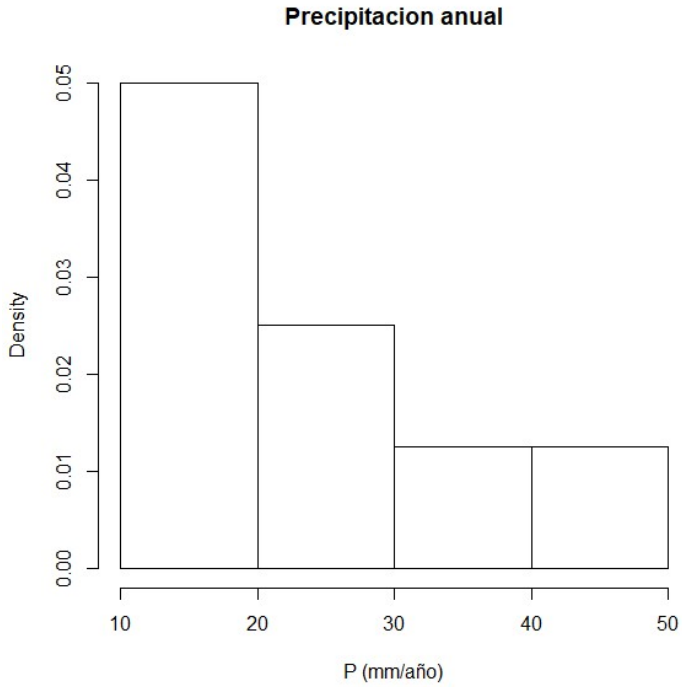
En la consola de resultados, efectuar las siguientes operaciones:

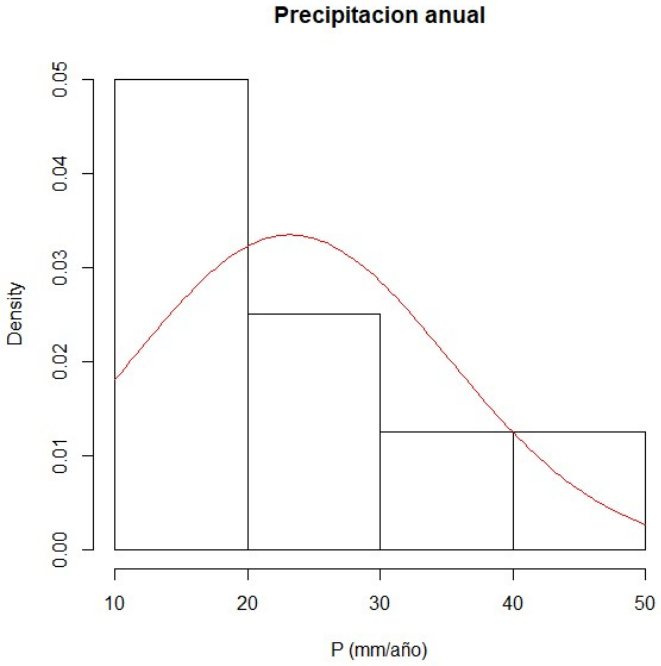
1) Plotear un histograma de frecuencias para rain, empleando la frecuencia
2) Plotear un histograma de frecuencias para rain, empleando la densidad
3) Agregar al histograma una curva de distribución normal en color rojo con la media y desviación estimados anteriormente
4) Agregar en un solo grafico (1filas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

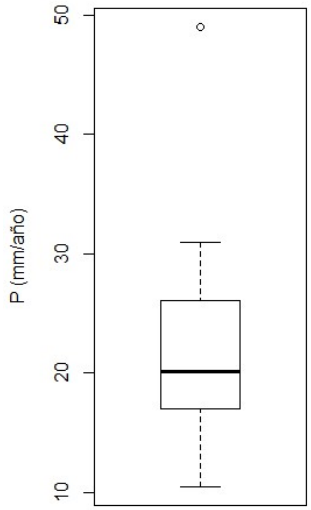
Respuestas A.3

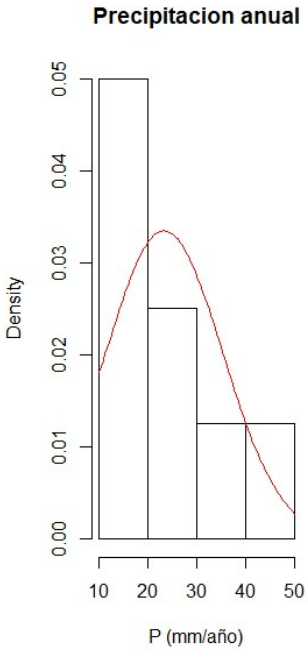
```
> hist(rain, main="Precipitacion anual", xlab="P (mm/año)", freq=F) # Grafico de histograma, F(False) no usar frecuencia
> curve(dnorm(x, mean(rain, na.rm = T), sd(rain, na.rm = T)), add = TRUE, col="red") # Agregar curva distribucion r normal en color rojo
>split.screen(c(1,2)) #usar este comando 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,soi) #covarianza
[1] 5.650357
> cor(rain,soi) #coeficiente de correlación r
[1] 0.8620822
> lm(soi ~ rain) #ecuación de regresión lineal
Call:
lm(formula = soi ~ rain)
Coefficients:
(Intercept)        rain
   -0.36867      0.03978
> plot(rain,soi,main="ENSO-rainfall relationship",xlab="rainfall", ylab="SOI index")
> abline(lm(soi~rain))
```

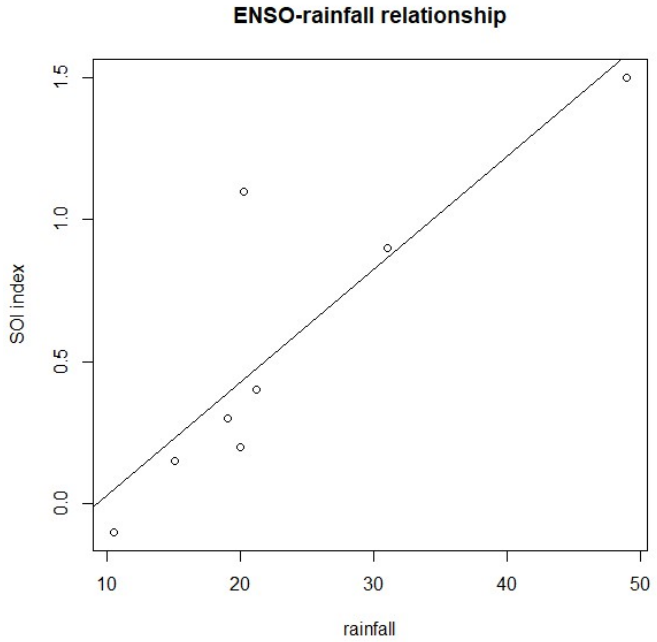

Precipitacion anual
Frequency
P (mm/año)


Precipitacion anual
Density
P (mm/año)


Precipitacion anual
Density
P (mm/año)


P (mm/año)


Precipitacion anual
Density
P (mm/año)


ENSO-rainfall relationship
SOI index
rainfall

Ejercicio A.4

En la consola de scripts:

1) Crear una función que calcule la velocidad de un rio ancho por el método de Manning: datos de ingreso el tirante, la pendiente y el coeficiente de Manning en el S.I.
2) Calcular la velocidad para para un tirante de 1.5m, pendiente de 1% y un coeficiente de Manning de 0.035.
3) Calcular la velocidad para un tirante de 1.5m, pendiente de 0.5% y un coeficiente de Manning de 0.035

Respuestas A.4

$$V = \frac{1}{n} R_h^{\frac{2}{3}} \cdot S^{\frac{1}{2}}$$

V: Velocidad
Rh: Radio Hidraulico
S: Pendiente
n: Coeficiente de Manning

```
vel.manning<- function(y,s,n)   #creando la funcion vel.manning
con tres argumentos
{ result <- y^(2/3)*s^0.5/n   #definiendo las operaciones
 print(result) }

vel.manning(1.5,0.01,0.035) # calculo de la velocidad en m/s
[1] 3.743916

vel.manning(1.5,0.005,0.035) # calculo de la velocidad en m/s
[1] 2.647349
```

3. Análisis exploratorio de datos

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, ...)

Leyendo archivos, bases de datos

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

Paradis (2002)

Configurando Series de tiempo

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

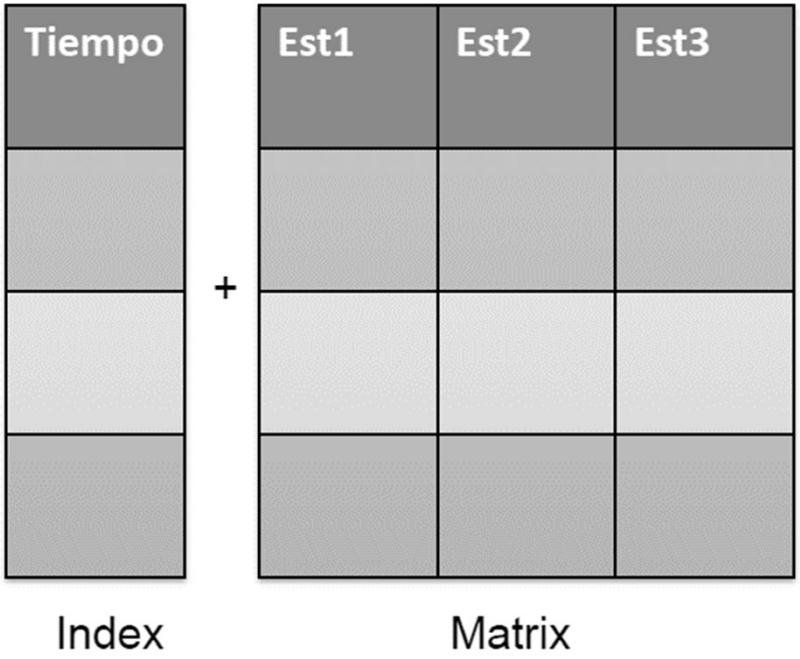
data	a vector or a matrix
start	the time of the rst observation, either a number, or a vector of two integers (see the examples below)
end	the time of the last observation specied in the same way than start
frequency	the number of observations per time unit
deltat	the fraction of the sampling period 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 dierence 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)



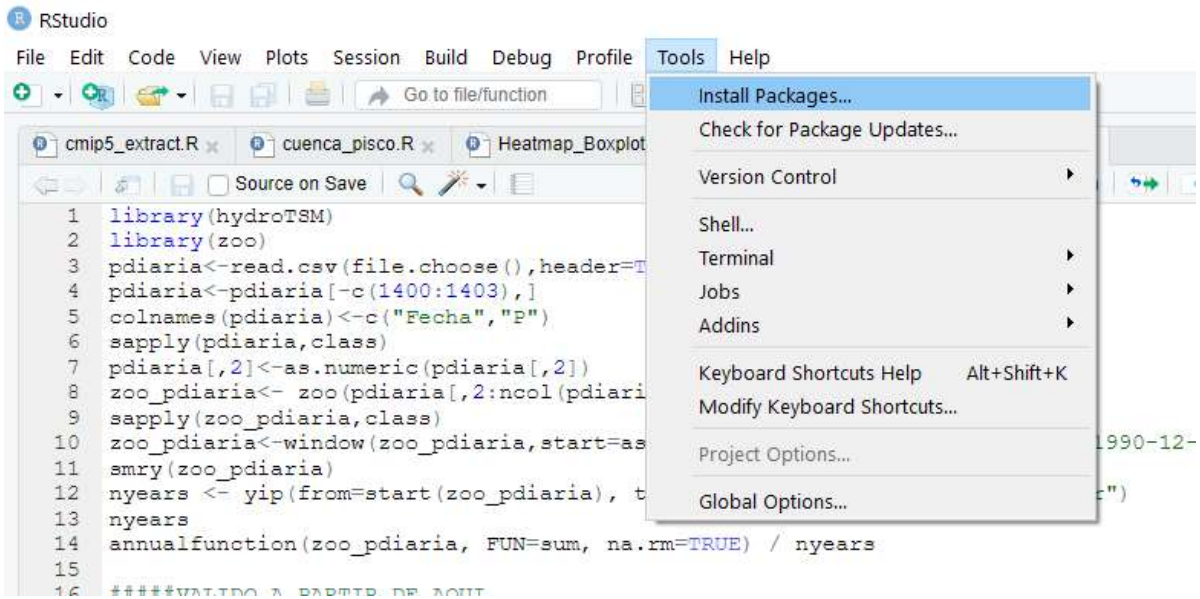
B. Explorando datos de precipitación mensual almacenados en una matriz de una sola estación

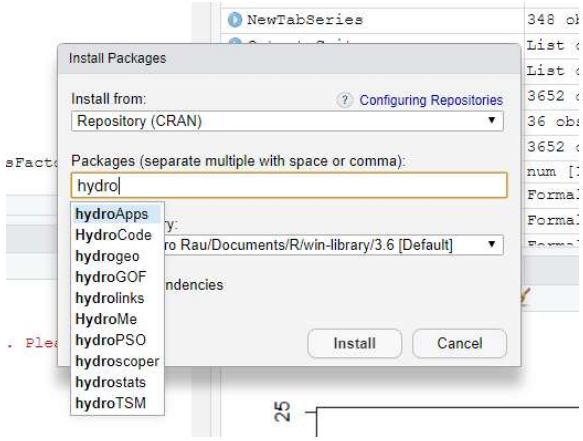
Uso de las librerías: lattice, zoo, hydroTSM

- Instalar librerías TOOLS - INSTALL PACKAGES: lattice, zoo, hydroTSM
- Descargar archivo "pmensual.txt"

https://drive.google.com/file/d/1Z1k6QOpFeyaOBvRyzgb5kH4Od8pLOpKS/view?usp=sharing

- Revisión de archivo texto (guardado desde el excel)





Ejercicio B

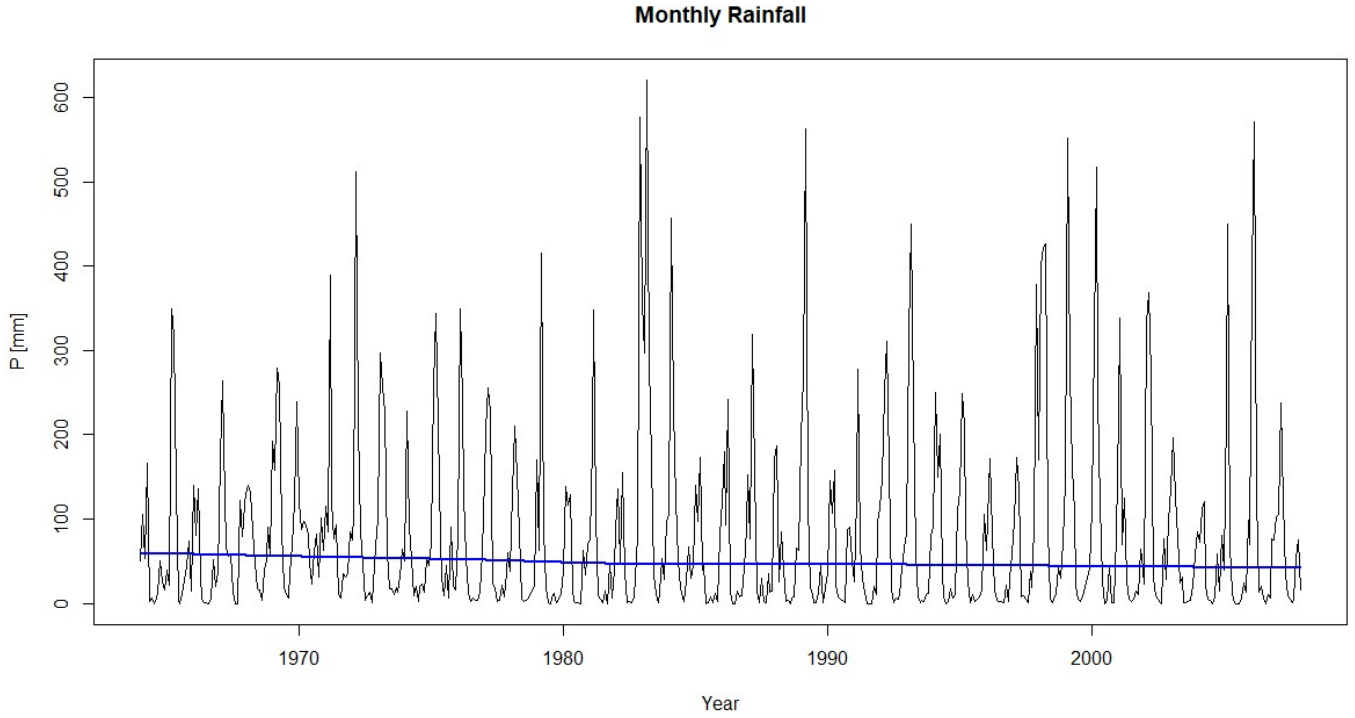
En la consola de scripts, crear un código que realice lo siguiente:

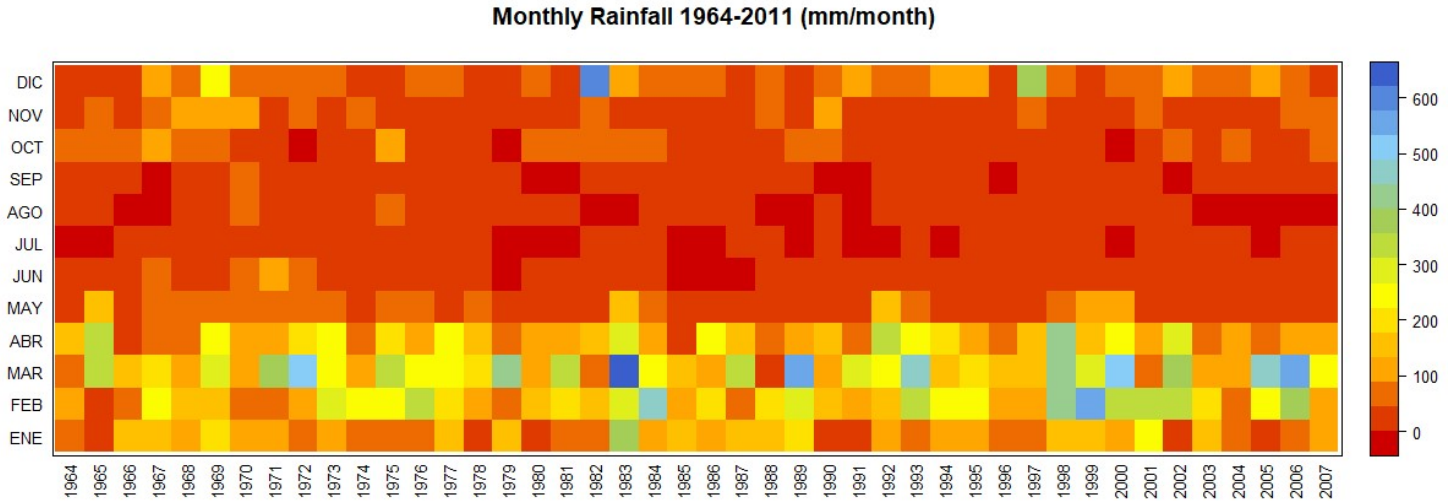
1) Leer el archivo de precipitaciones mensuales "pmensual.txt" almacenados en formato matriz de 13 columnas (año mes1 mes 2 …. mes 12) y 45 filas (nombre valor1 valor 2 …).
2) Plotear toda la serie de tiempo
3) Agregar una curva de tendencia
4) Plotear un gráfico de calor "heatmap"

Grabar y dar nombre al script: Analisis_mensual.R

Respuestas B

```
library("lattice") #llamar librería lattice
library("hydroTSM") #llamar librería TSM
pmensual<-read.table(file.choose(), header = F) #lectura de archivo, se abrirá una ventana para seleccionar el archivo, cual fuese su ruta, header "F" por no contener datos en la cabecera solo contiene etiquetas

#Visualización de la serie de tiempo:
datos<-pmensual[2:45,2:13]
datos_vector<-as.vector(t(datos)) #convirtiendo a vector lineal
datos_ts<-stats::ts(datos_vector, start=c(1964, 1), end=c(2010, 12), frequency=12)
plot.ts(datos_ts, col="black", main="Monthly Rainfall time series", ylab="P [mm]", xlab="Year")
lines(lowess(time(datos_ts), datos_ts), col="blue", lwd=2) #agregar curva de tendencia

#Diagrama de calor o HeatMap:
lluvia<-pmensual[2:45,2:13] #lectura de solo datos de lluvia, sin etiquetas de años, ni meses (desde 1964 al 2007)
meses<-pmensual[1:1,2:13] #lectura de la cabecera de meses en la primera fila
colnames(lluvia)<-unlist(meses) #desagrega los nombres de los meses y se asigna a la matriz
rownames(lluvia)<-pmensual[2:45,1:1] #desagrega los nombres de los meses y se asigna a la matriz
matrixplot(lluvia, ColorRamp="Precipitation",main="Monthly Rainfall 1964-2011 (mm/month)")
```


Monthly Rainfall
P [mm]
0
100
200
300
400
500
600
1970
1980
1990
2000
Year


Monthly Rainfall 1964-2011 (mm/month)
DIC
NOV
OCT
SEP
AGO
JUL
JUN
MAY
ABR
MAR
FEB
ENE
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
600
500
400
300
200
100
0

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

Uso de las librerias: easypackages, xts, lattice, ggplot2

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

-Descargar archivo "p_diarias.csv"

https://drive.google.com/file/d/1qj6I51SjhNIUfhq56Fw8kK4v3g1ttYq8/view?usp=sharing

-Revisión de archivo csv (separados por comas con formato de fecha %Y-%M-%D)

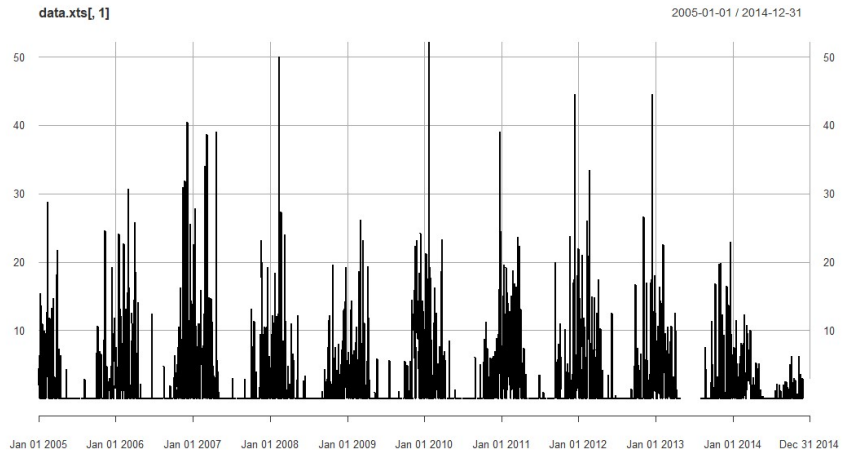
Ejercicio C

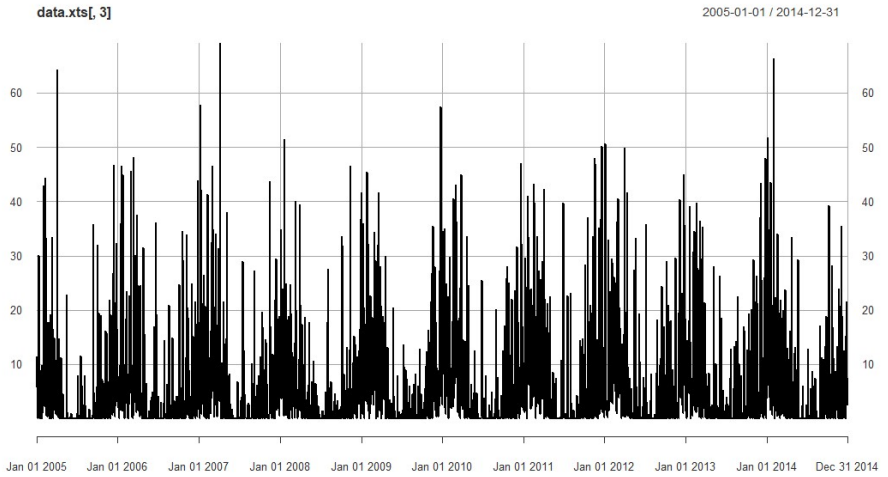
En la consola de scripts, crear un código que realice lo siguiente:

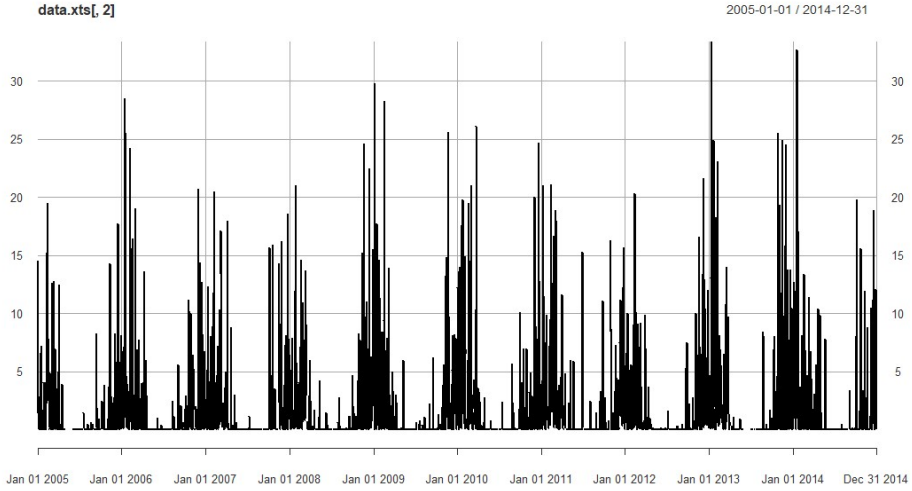
1) Leer el archivo de precipitaciones diarias "p_diarias.csv" almacenados en formato matriz con 4 estaciones (identificadores 101, 102, 103, 104) desde el 1Ene2005 al 31Dic2014
2) Plotear las series de tiempo para cada estación
3) Plotear las series de tiempo en conjunto con la misma escala vertical
4) Plotear un boxplot de las estaciones en conjunto
5) Plotear 3 histogramas correspondientes a 3 estaciones

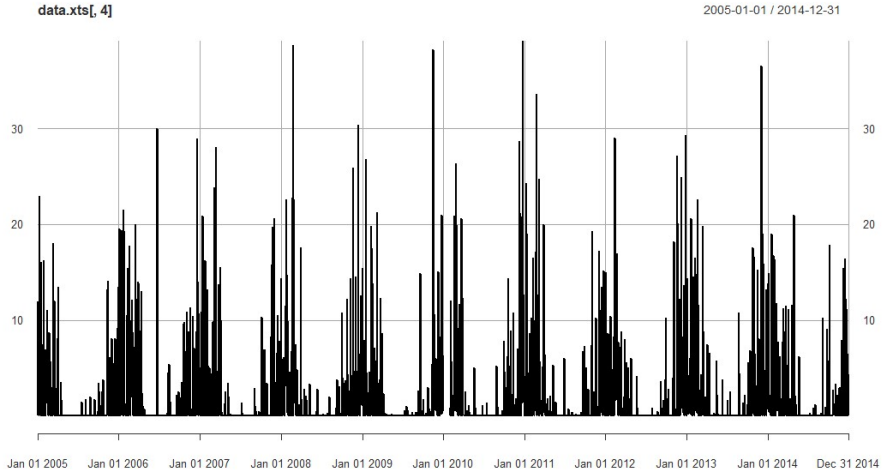
Respuestas C

```
library(easypackages)
library(xts)
library(lattice)
library(ggplot2)
pdiaria <- read.csv(file.choose(),header=TRUE, check.names = F, stringsAsFactors = F)
str(pdiaria) #para verificar 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 que almacene varias estaciones a la vez
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))
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
```

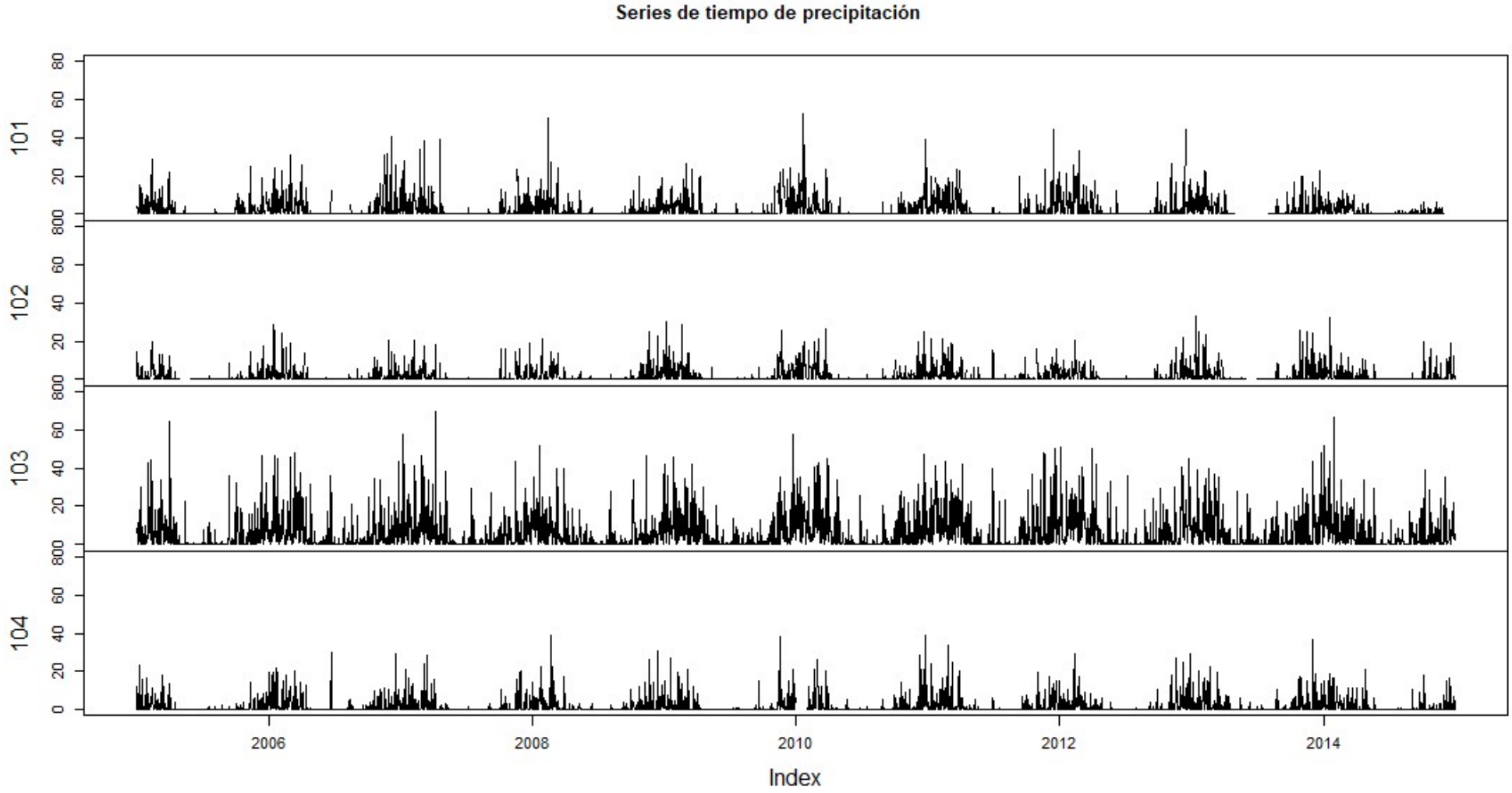

data.xts[, 1]
2005-01-01 / 2014-12-31
50
40
30
20
10
Jan 01 2005
Jan 01 2006
Jan 01 2007
Jan 01 2008
Jan 01 2009
Jan 01 2010
Jan 01 2011
Jan 01 2012
Jan 01 2013
Jan 01 2014
Dec 31 2014


data.xts[, 3]
2005-01-01 / 2014-12-31
60
50
40
30
20
10
Jan 01 2005
Jan 01 2006
Jan 01 2007
Jan 01 2008
Jan 01 2009
Jan 01 2010
Jan 01 2011
Jan 01 2012
Jan 01 2013
Jan 01 2014
Dec 31 2014

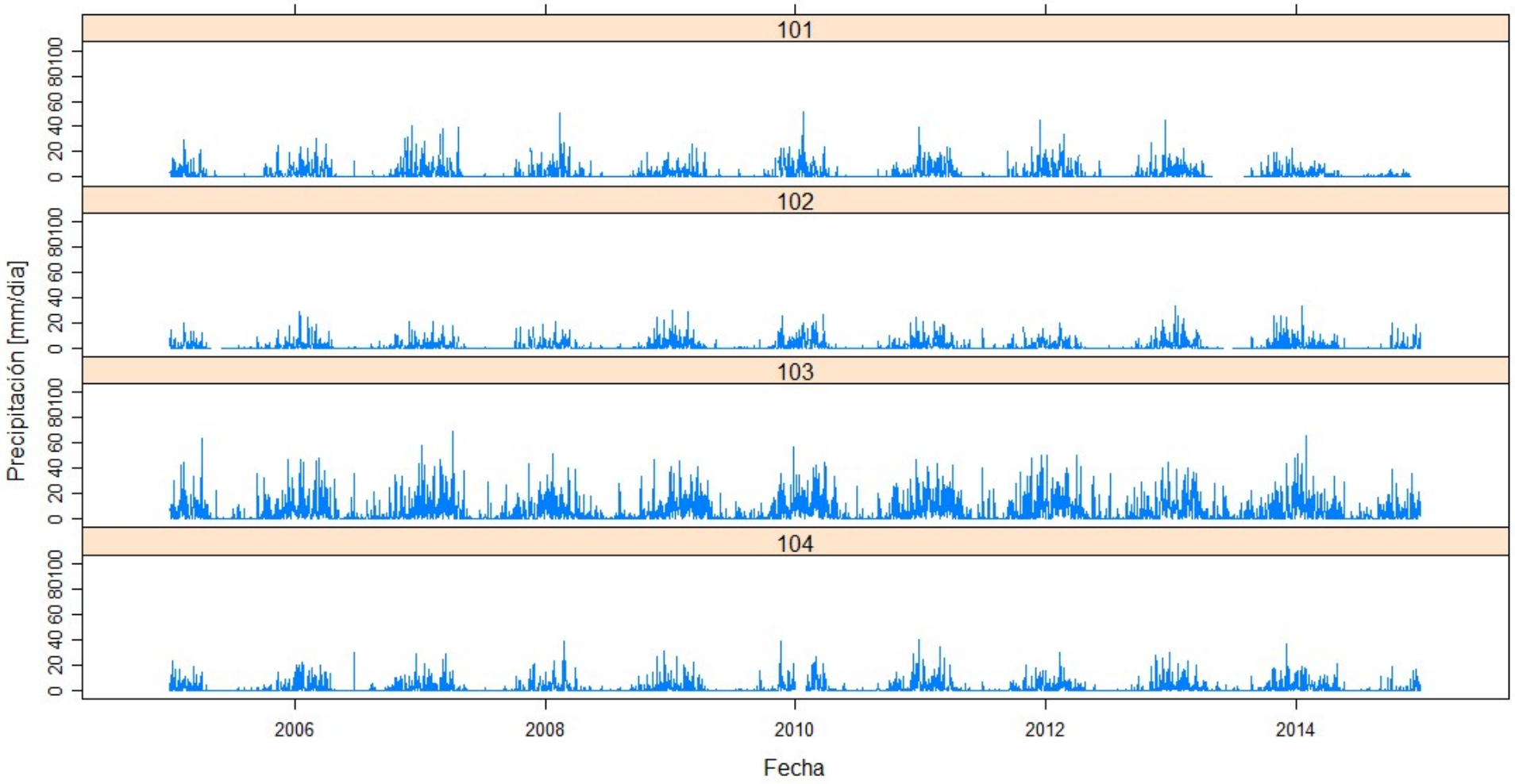

data.xts[, 2]
2005-01-01 / 2014-12-31
30
25
20
15
10
5
Jan 01 2005
Jan 01 2006
Jan 01 2007
Jan 01 2008
Jan 01 2009
Jan 01 2010
Jan 01 2011
Jan 01 2012
Jan 01 2013
Jan 01 2014
Dec 31 2014


data.xts[, 4]
2005-01-01 / 2014-12-31
30
20
10
Jan 01 2005
Jan 01 2006
Jan 01 2007
Jan 01 2008
Jan 01 2009
Jan 01 2010
Jan 01 2011
Jan 01 2012
Jan 01 2013
Jan 01 2014
Dec 31 2014

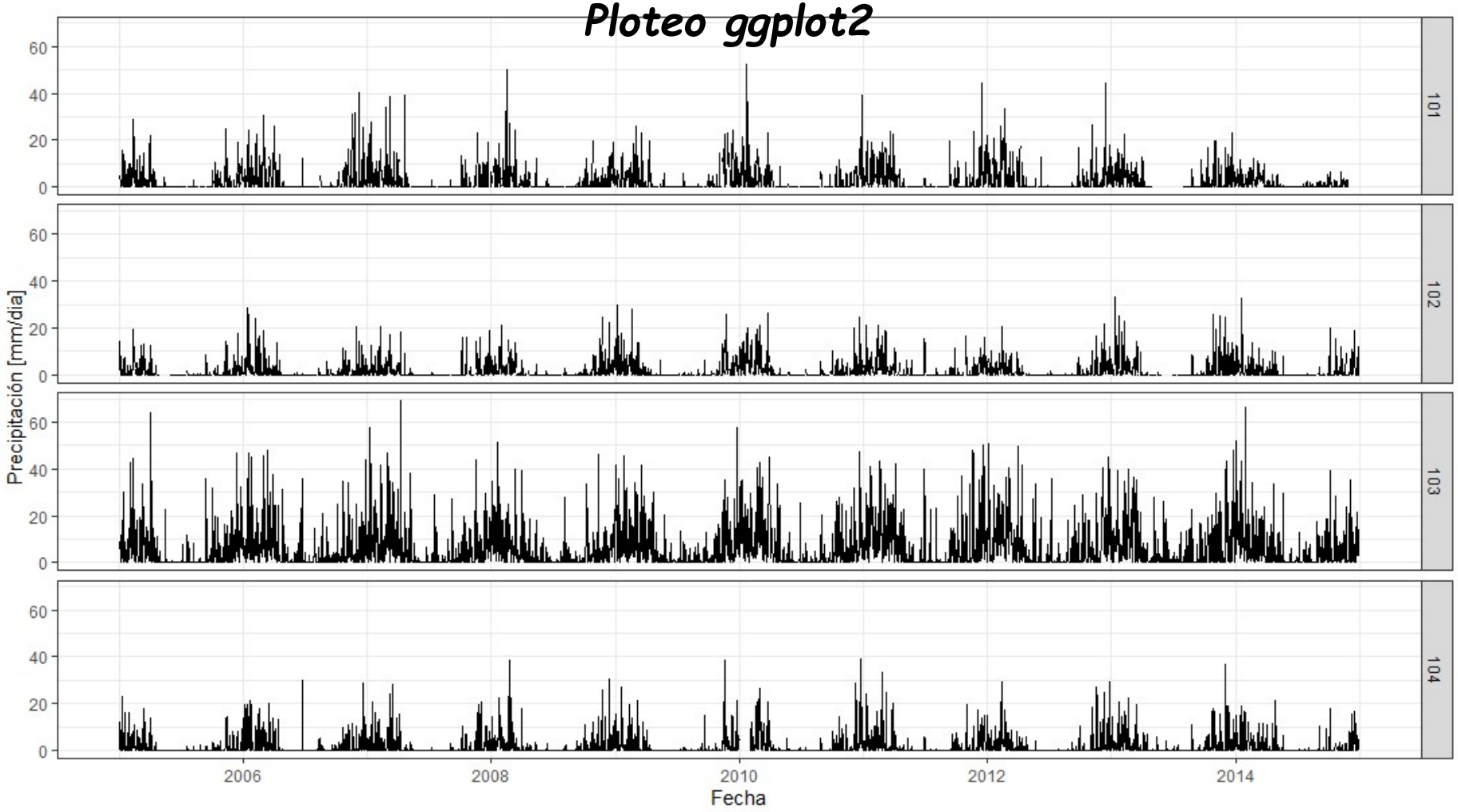
Ploteo simple

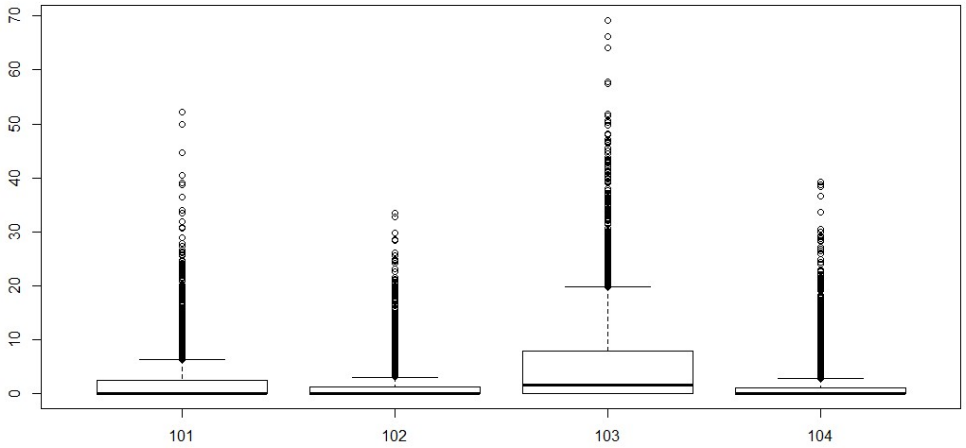


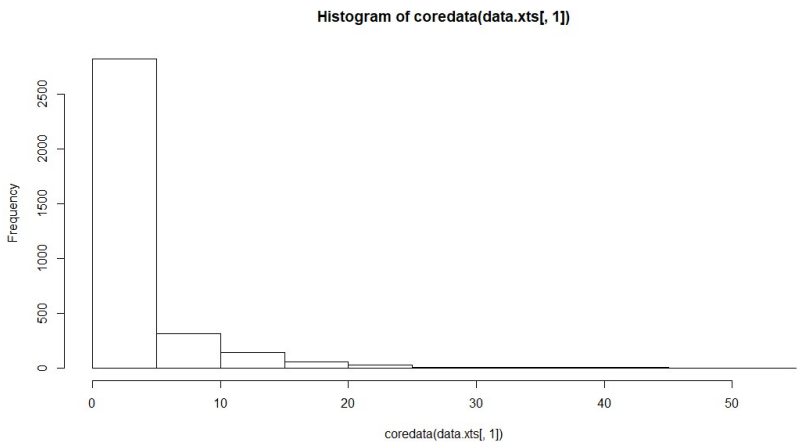
Ploteo lattice

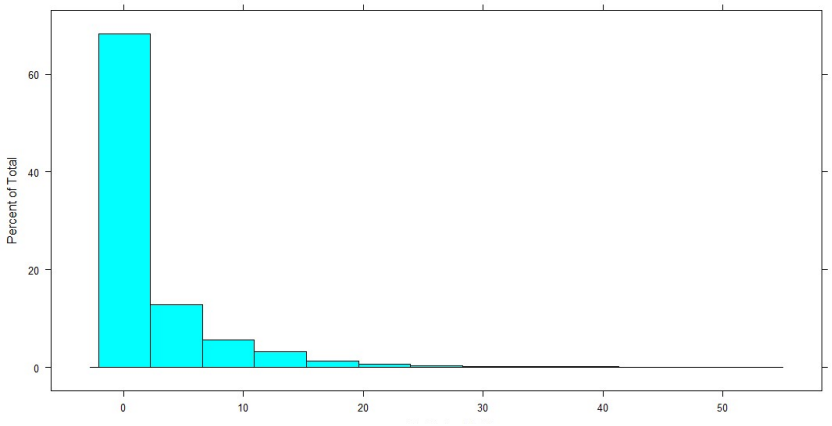


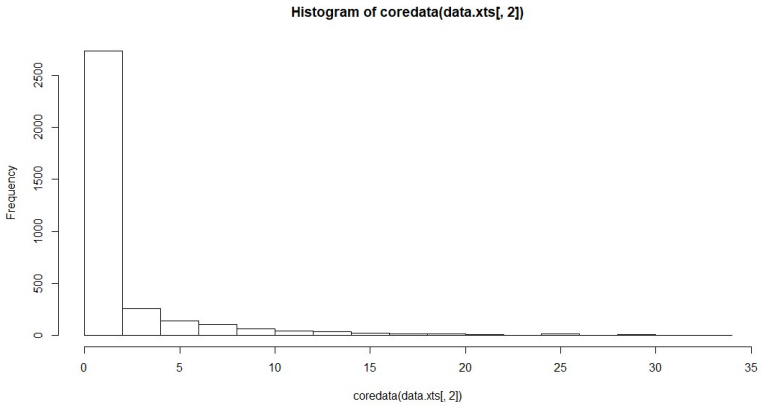
Ploteo ggplot2

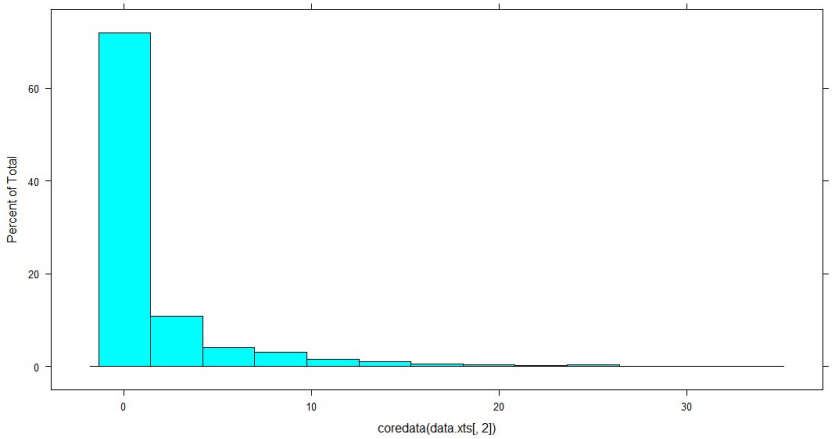


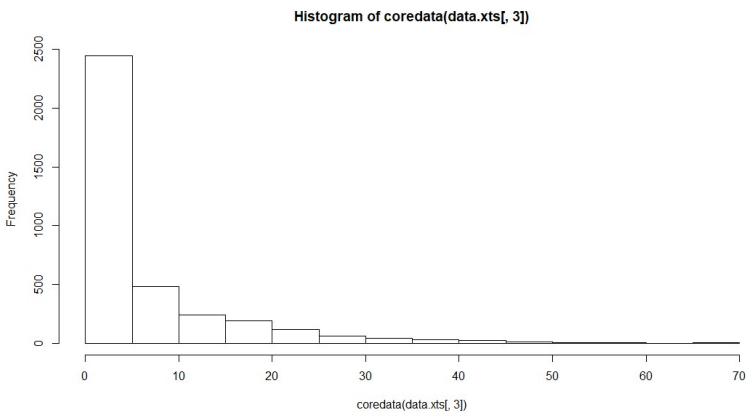


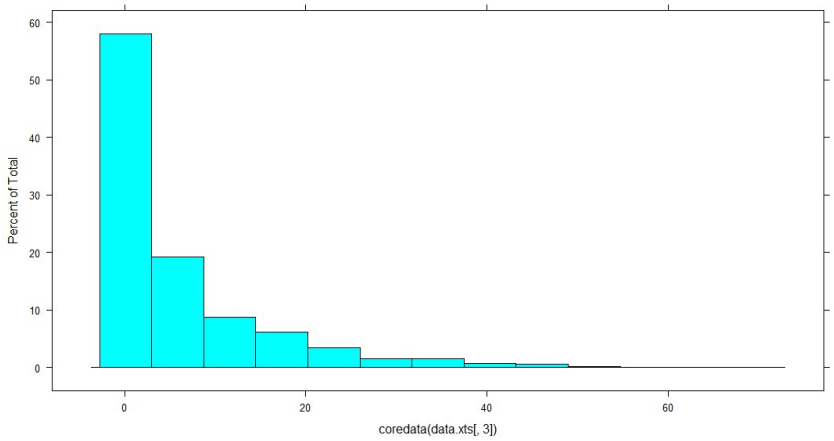

Histogram of coredata(data.xts[, 1])
Frequency
0
500
1000
1500
2000
2500
0
10
20
30
40
50
coredata(data.xts[, 1])


Percent of Total
0
20
40
60
0
10
20
30
40
50


Histogram of coredata(data.xts[, 2])
Frequency
0
500
1000
1500
2000
2500
0
5
10
15
20
25
30
35
coredata(data.xts[, 2])


Percent of Total
0
20
40
60
0
10
20
30
coredata(data.xts[, 2])


Histogram of coredata(data.xts[, 3])
Frequency
0
500
1000
1500
2000
2500
0
10
20
30
40
50
60
70
coredata(data.xts[, 3])


Percent of Total
0
10
20
30
40
50
60
0
20
40
60
coredata(data.xts[, 3])

D. Convirtiendo datos diarios a mensuales y anuales

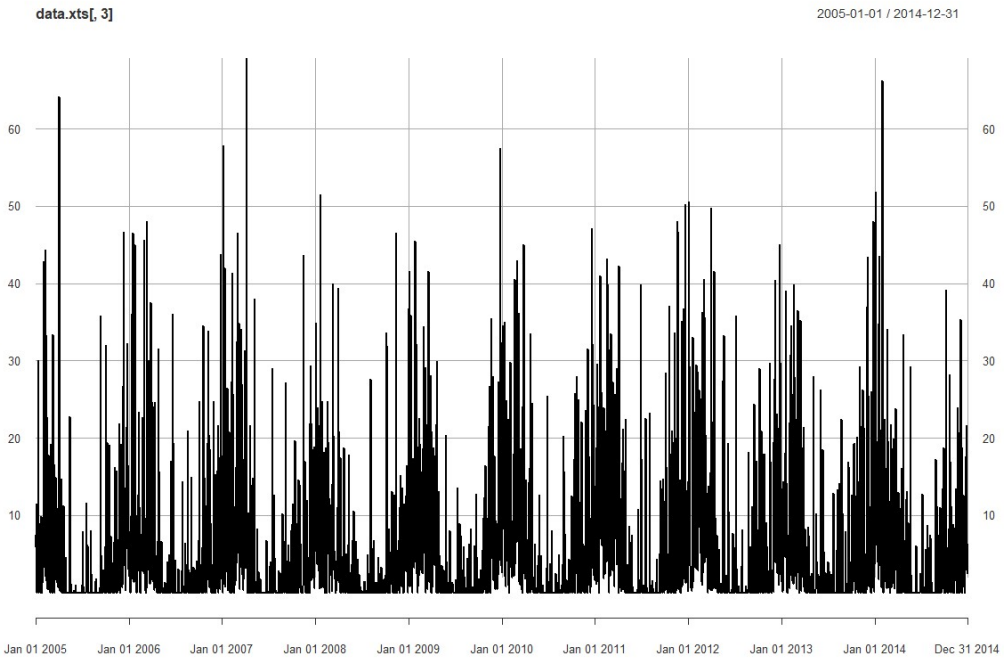
Ejercicio D

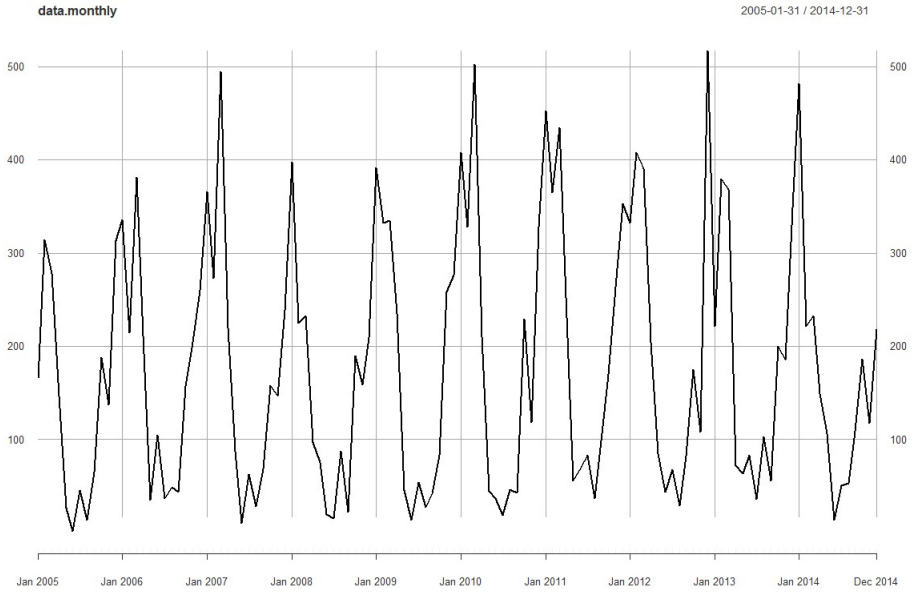
En la consola de scripts, crear un código que realice lo siguiente:

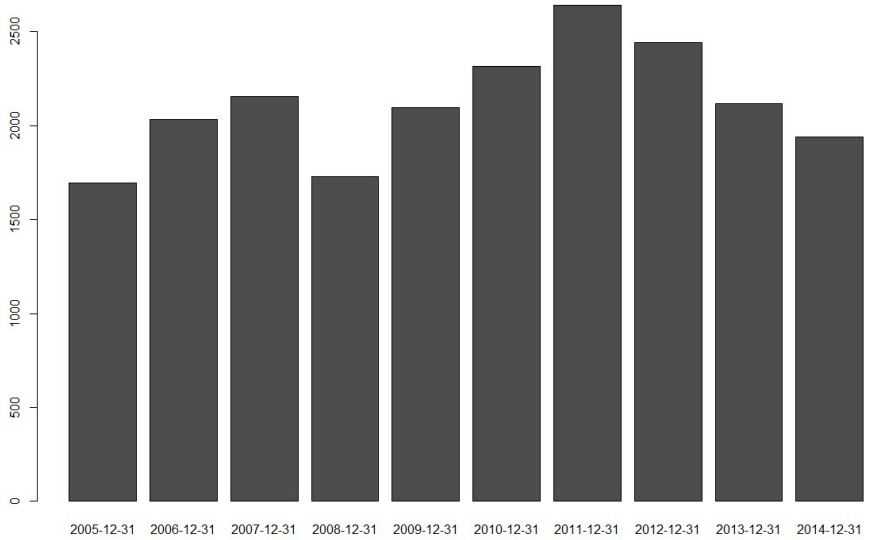
1) Leer el archivo de precipitaciones diarias "p_diarias.csv" almacenados en formato matriz con 4 estaciones (identificadores 101, 102, 103, 104) desde el 1Ene2005 al 31Dic2014
2) La estación 103 se encuentra completa, convertir sus datos diarios en mensuales y éstos en anuales. Plotear las series de tiempo, los datos mensuales en formato línea y los anuales en formato barras. Plotear todas las estaciones en conjunto y analizar el vacío de información.
3) Plotear un boxplot para los datos mensuales de cada estación
4) Plotear un boxplot estacional de la estación 103

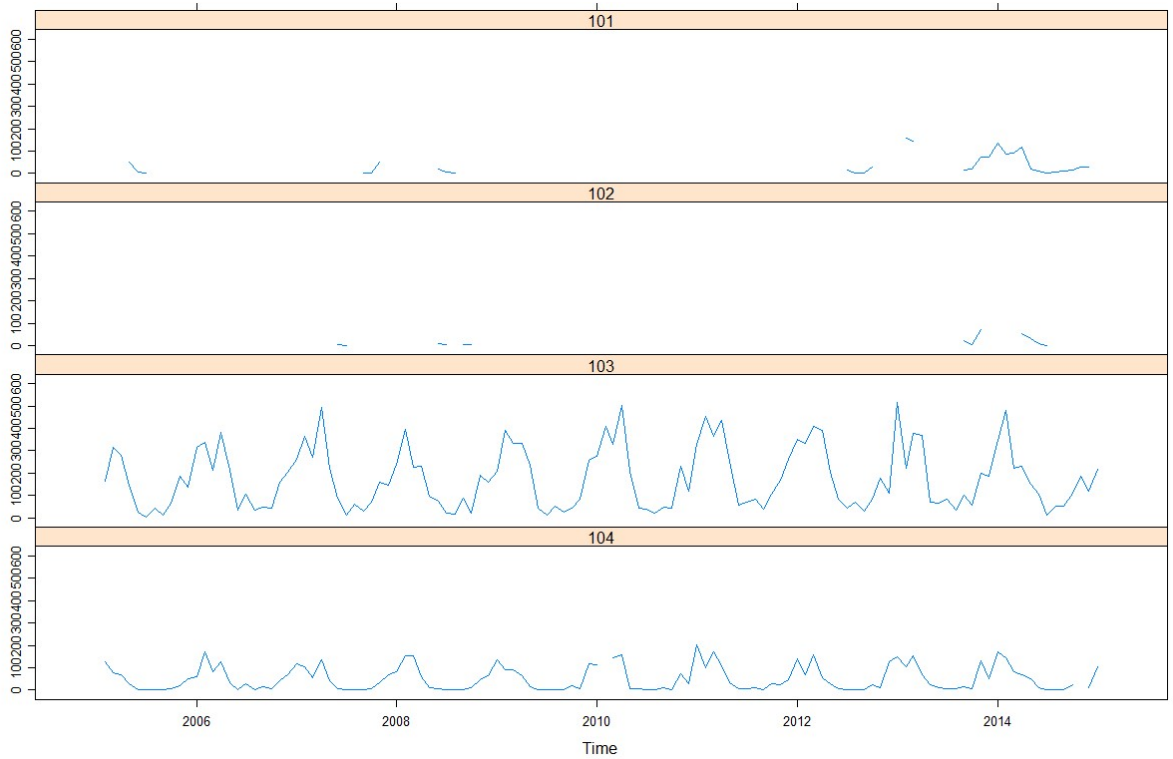
Respuestas D

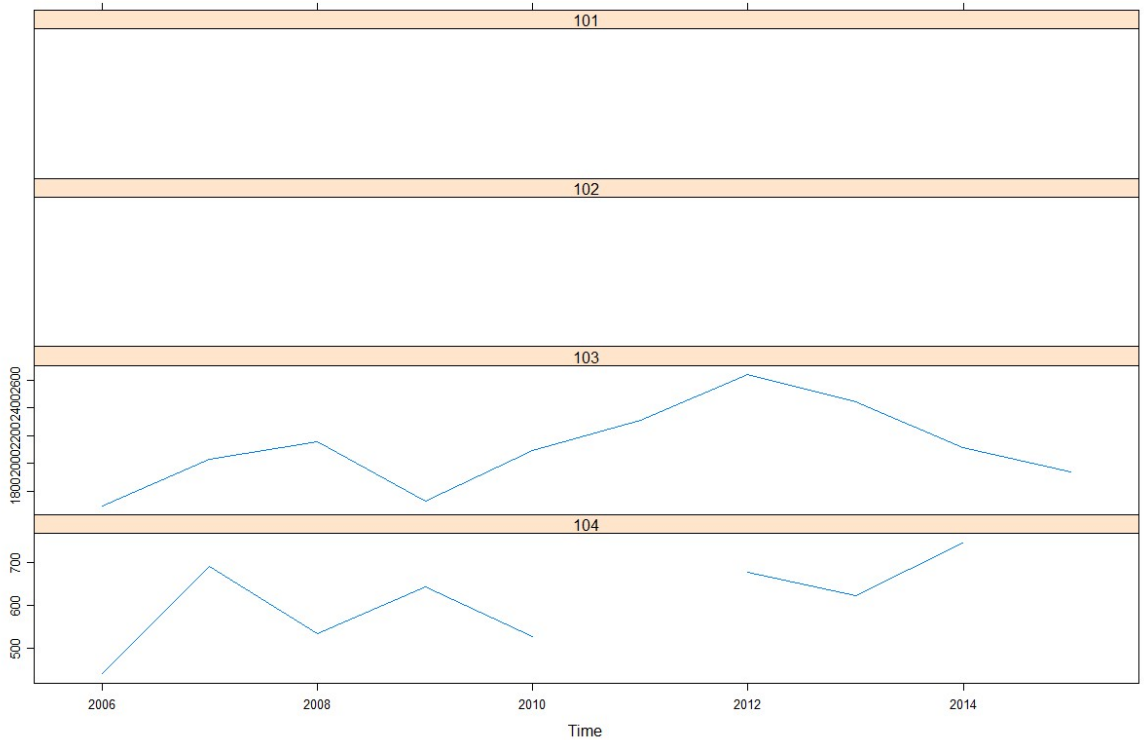
```
library(easypackages)
library(xts)
library(lattice)
library(ggplot2)
pdiaria <- read.csv(file.choose(),header=TRUE, check.names = F, stringsAsFactors = F)
str(pdiaria)
idx <- as.Date(pdiaria[,1])
data.matrix <- pdiaria[,-1]
data.xts <- xts(data.matrix, order.by = idx )
str(data.xts)
plot(data.xts)
plot(data.matrix[,1], type="l")
plot(data.xts[,3])
#Convirtiendo a mensuales la estacion 103
data.monthly <- apply.monthly(data.xts[,3], FUN = sum)
plot(data.monthly)
#Todas las estaciones a mensuales
data.monthly <- apply.monthly(data.xts, FUN=apply, MARGIN = 2, sum)
xyplot(data.monthly, ylim = c(0,600))
#Convirtiendo a anuales la estacion 103
data.anual <- apply.yearly(data.xts[,3], FUN = sum)
barplot(data.anual)
#Todas las estaciones a anuales
data.anual <- apply.yearly(data.monthly, FUN=apply, 2, sum)
xyplot(data.anual)

boxplot(coredata(data.monthly)) # datos mensuales por estacion
# boxplot con estacionalidad de Lluvia para la estacion 103
boxplot(matrix(coredata(data.monthly[,3]), nrow=nrow(data.monthly)/12, ncol=12, byrow=T),
col="gray", main=c(paste(names(data.monthly[,3])), "Prec mensual [mm]"))
```

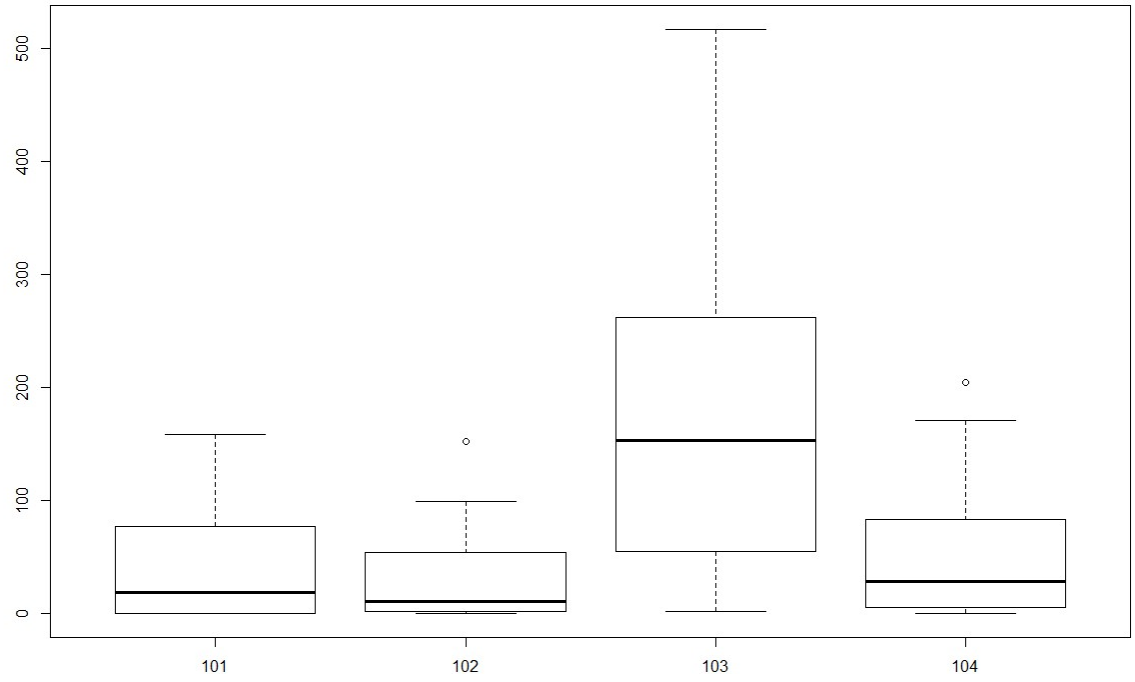

data.xts[, 3]
2005-01-01 / 2014-12-31
60
50
40
30
20
10
Jan 01 2005
Jan 01 2006
Jan 01 2007
Jan 01 2008
Jan 01 2009
Jan 01 2010
Jan 01 2011
Jan 01 2012
Jan 01 2013
Jan 01 2014
Dec 31 2014


data.monthly
2005-01-31 / 2014-12-31
500
400
300
200
100
Jan 2005
Jan 2006
Jan 2007
Jan 2008
Jan 2009
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Jan 2014
Dec 2014

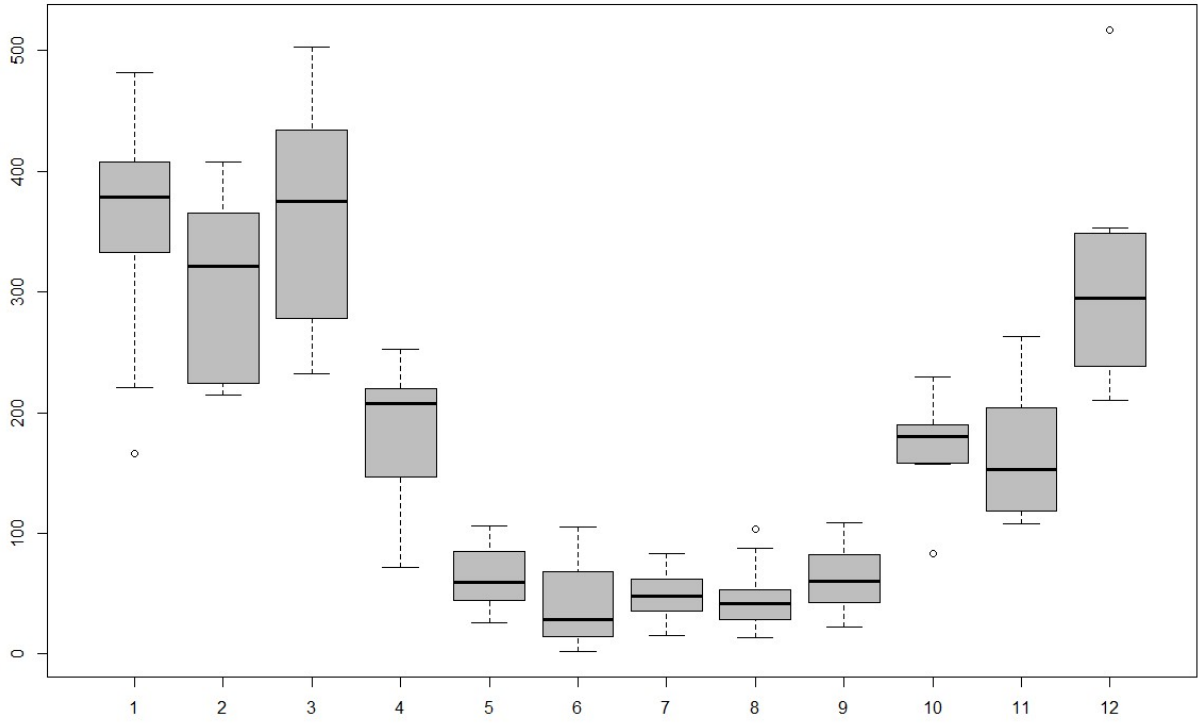

2500
2000
1500
1000
500
0
2005-12-31
2006-12-31
2007-12-31
2008-12-31
2009-12-31
2010-12-31
2011-12-31
2012-12-31
2013-12-31
2014-12-31


101
102
103
104
0 100200300400500600
2006
2008
2010
2012
2014
Time


101
102
103
104
1800200022002400 2600
500 600 700
2006
2008
2010
2012
2014
Time



103
Prec mensual [mm]



Anexo - Beyond the library

ggplot2

ggplot(data, ...) + CAPA 1 + CAPA 2 + CAPA 3 + ...

Geometría:
+ geom_point( )
+ geom_text( )
+ geom_line( )
+ geom_area( )
+ geom_density( )
+ geom_histogram( )
+ geom_bar( )
+ geom_boxplot( )

Escalas:
+ scale_x_date( )
+ scale_x_datetime( )
+ scale_x_log10( )
+ scale_x_reverse( )

Etiquetas:
+ ggtitle( )
+ xlab( )
+ ylab( )

Otros:
+ stat_ecdf

Temas:
+ theme_bw( )
+ theme_classic( )
+ theme_minimal( )

Package
Ggthemes
Más temas

ggplot2

La **data** ingresada siempre es un **data frame !**

Ggplot (data, aes()) + capa1 (aes()) + …

Opción 1: **global**

Opción 2: **local**

Se deben llamar los elementos a usar (campos del data frame) mediante la función **aes()**

Múltiples gráficos ?

facet_grid(. ~ fl)	*en columnas*
facet_grid(year ~ .)	*en filas*
facet_grid(year ~ fl)	*en filas y columnas*
facet_wrap(~ fl)	*en arreglo rectangular*

ggplot2

Ejemplo de aplicación

Base de datos de precipitaciones diarias de estaciones de la Cuenca de Pativilca (2000-2007)

setwd(..***directorio***..)

load(dfPr.Rdata)

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'evolution. 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

Salas J. 1996. Analysis and modelling of hydrologic time series (in Handbook of Hydrology). McGraw-Hill Education.

Shen, C. 2018. Deep learning: A next-generation big-data approach for hydrology, Eos, 99

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

Wickham H, Grolemund G. 2016. R for data science. O'Reilly Media, Inc.

Wickham H. 2016. Ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York.