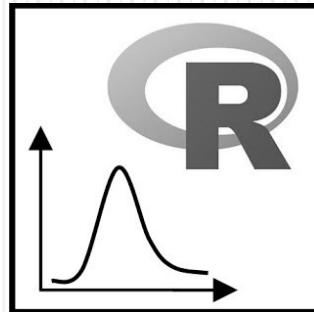


#useR #Rstudio
#hydrology #timeseries

Curso: Hidrología
Course: Hydrology

HidRología con R HydRology with R

Pedro Rau, PhD
Hidrólogo



<https://github.com/hydrocodes>



CENTRO DE
INVESTIGACIÓN
Y TECNOLOGÍA
DEL AGUA - UTEC

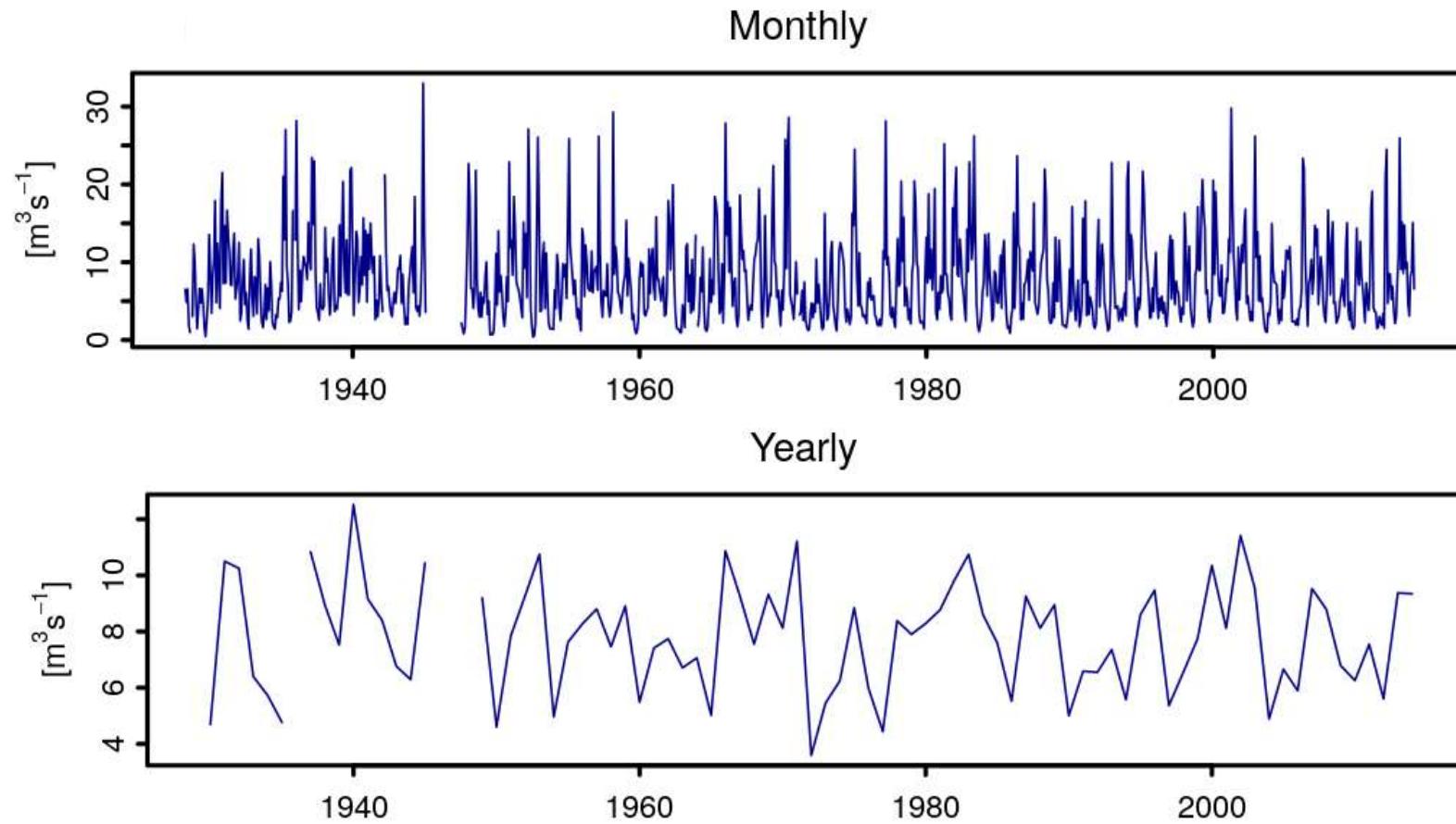
ESCUEDA DE
POSGRADO
UTEC



1. Introducción a las series de tiempo hidrológicas



1.1 ¿Cómo analizar una serie de tiempo?

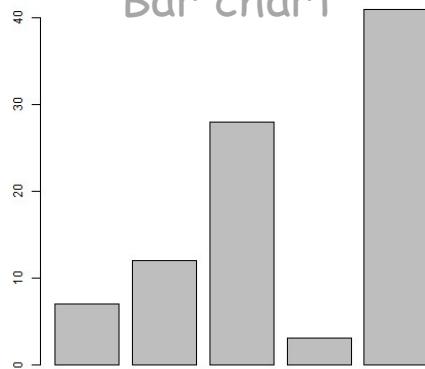


1.2 Análisis preliminar de datos hidrológicos

a. Algunos tipos de representaciones gráficas

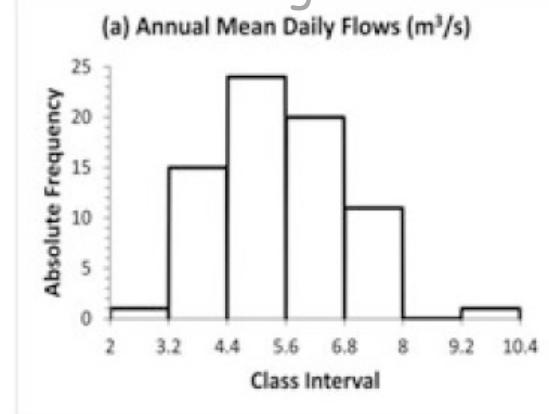
Diagrama de barras

Bar chart



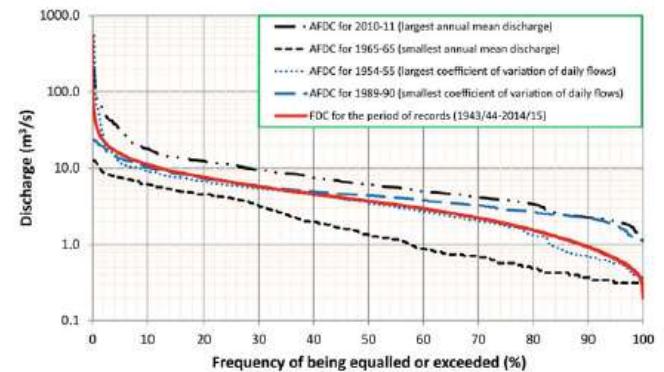
Histograma

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

$$x \text{ tal que } F(x) = 0.5$$

Estadística de la muestra

Valor de la información en el 50o. percentil

Coefficiente 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

Coefficiente de asimetría (oblicuidad)

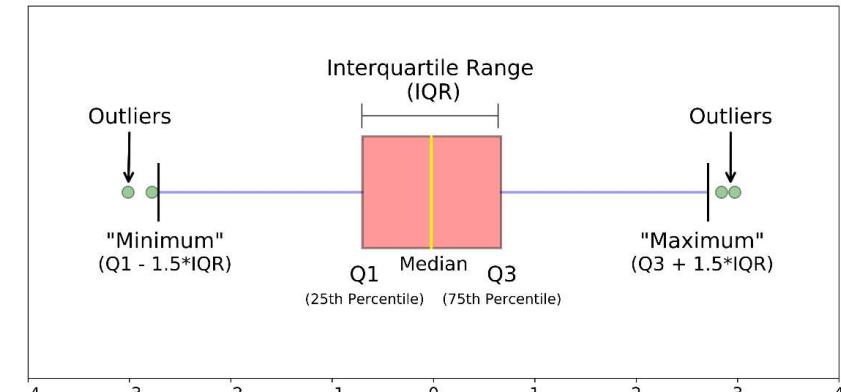
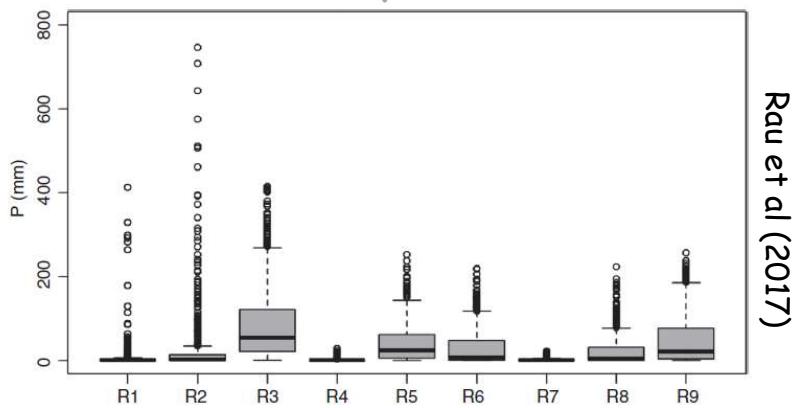
$$\gamma = \frac{E[(x - \mu)^3]}{\sigma^3}$$

$$C_s = \frac{n \sum_{i=1}^n (x_i - \bar{x})^3}{(n-1)(n-2)s^3}$$

b. Estadísticos descriptivos

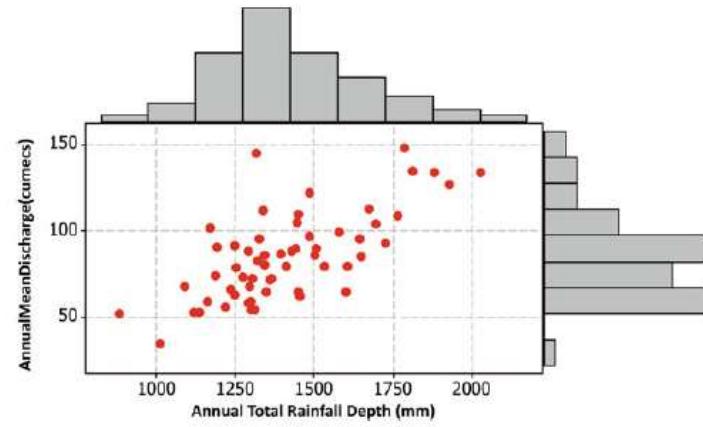
c. Algunos métodos exploratorios

Diagrama de cajas
Box plot



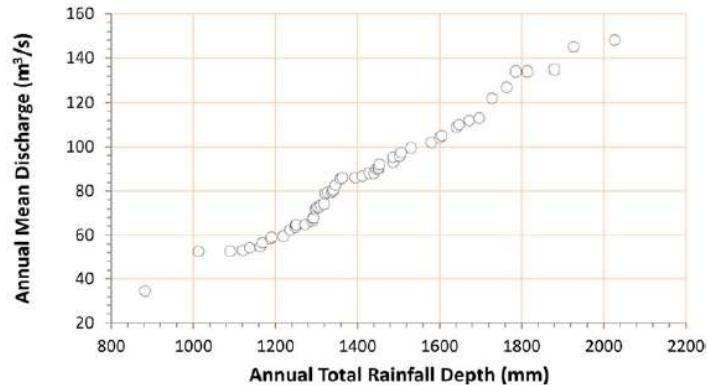
d. Asociación de datos

Gráfico de dispersion
Scatter plot



P. Rau –hidRología (2022)

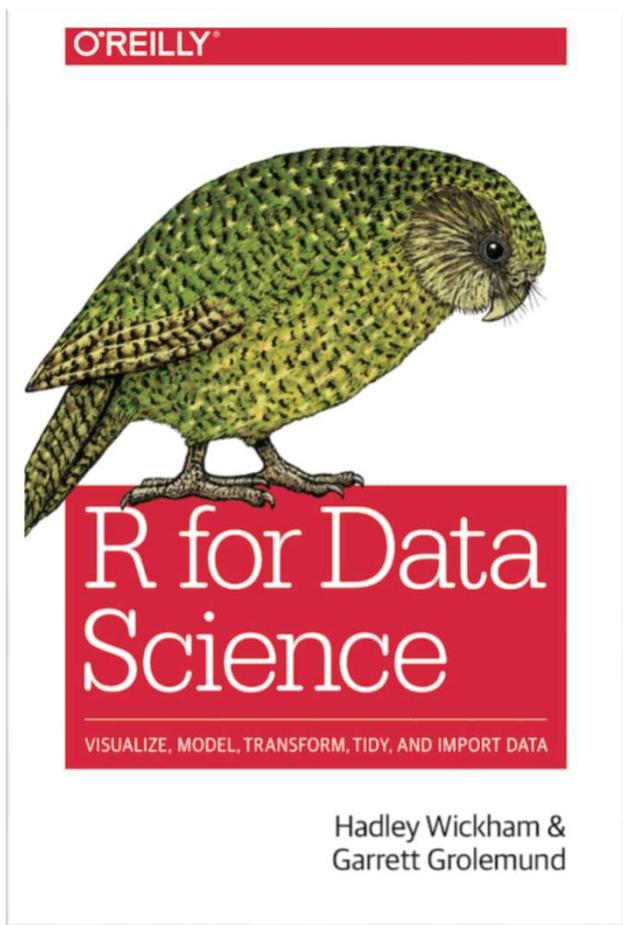
Diagrama Cuantil-Cuantil
Empirical Quantile-Quantile Diagram
Q-Q Plot





2. Entorno R y Rstudio

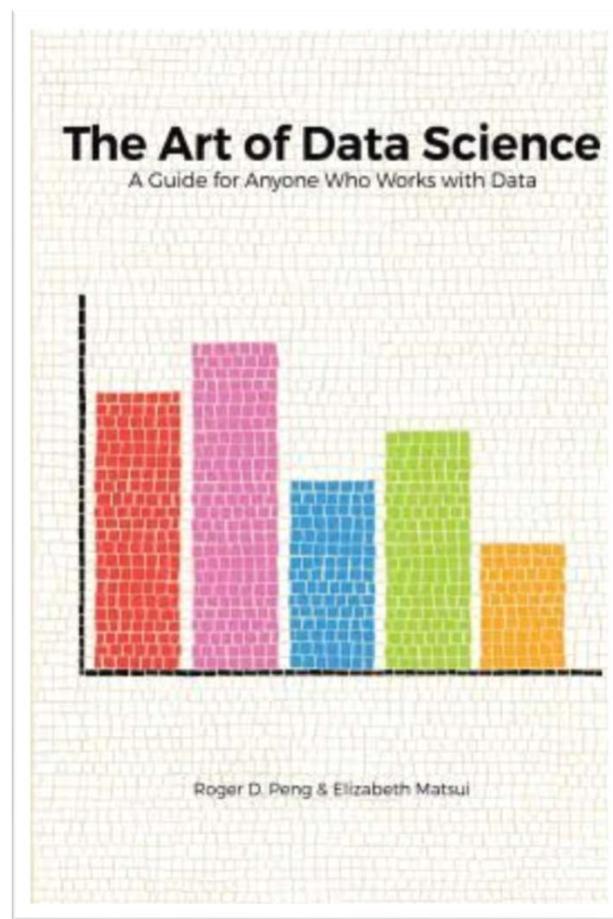




<https://r4ds.had.co.nz/>

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

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



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

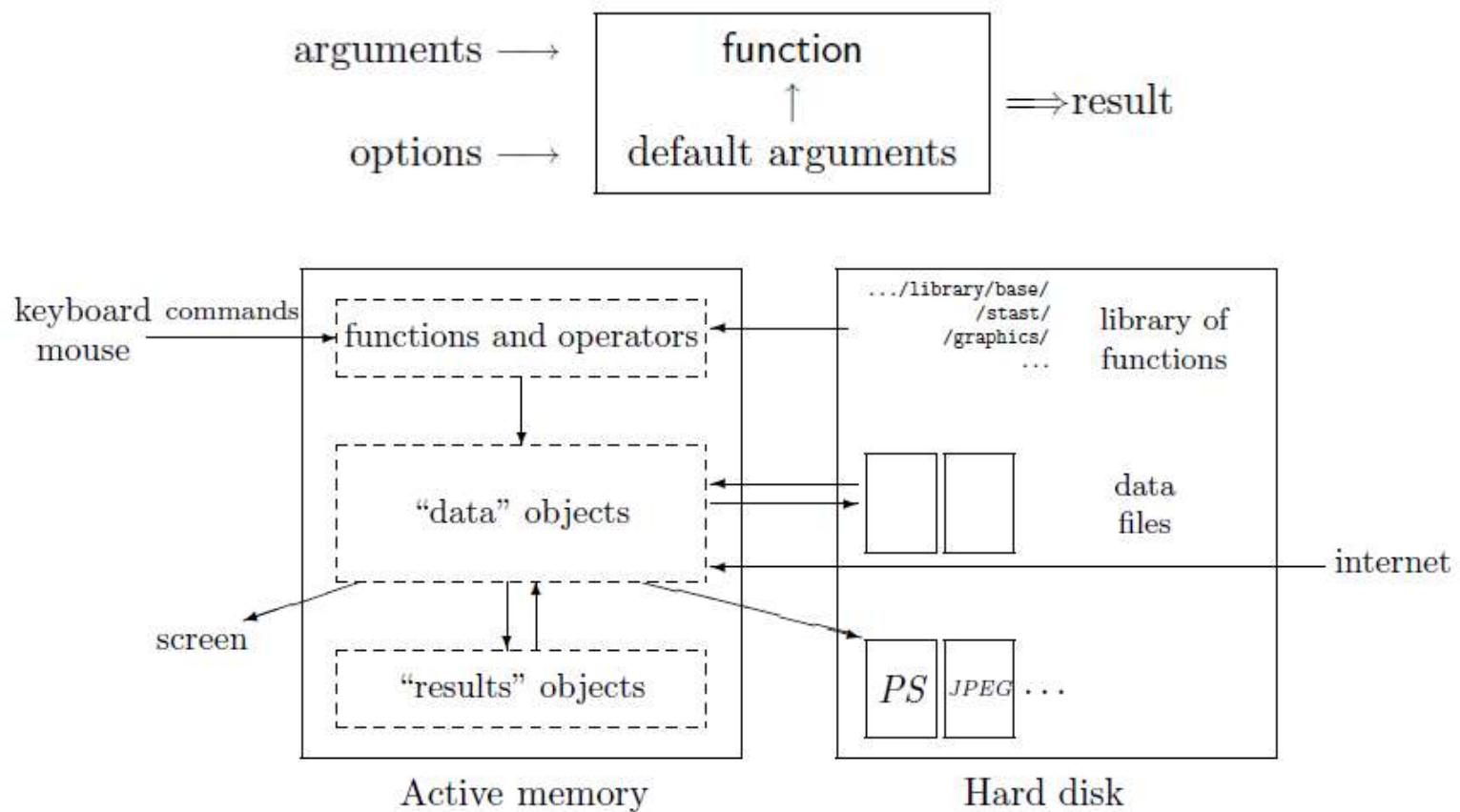
<https://github.com/waldronlab/The-Art-of-Data-Science/blob/master/README.md>

R-4.1.3 for Windows (32/64 bit) - Marzo 2022



<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»

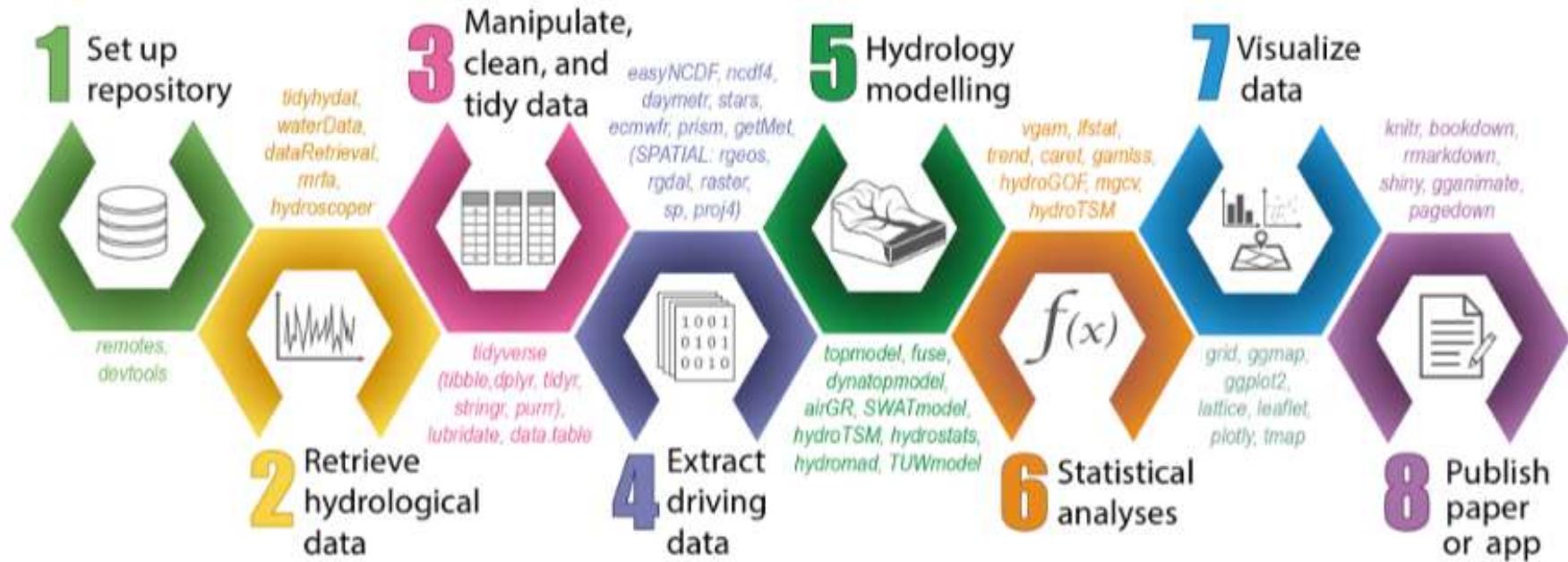


Paradis (2002)

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 (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	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 2022

<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. On the left, there is a code editor window titled 'A1.R' containing R script code. A red box highlights the top portion of the code editor and the bottom portion of the console, with the labels 'Editor de código' and 'Consola de scripts'. Below the code editor is a 'Console' tab in the console window, which contains R command history and output. A red box highlights this area with the labels 'Consola R' and 'Consola de resultados'. To the right of the code editor is the 'Environment' browser, showing a list of objects in the global environment. A red box highlights this area with the label 'Área de objetos e historial'. At the bottom right is a 'Plots' window displaying a line graph of discharge over time. A red box highlights this area with the label 'Figuras y archivos'.

Editor de código
Consola de scripts

Consola R
Consola de resultados

Área de objetos e historial

Figuras y archivos

```
1 library(devtools)
2 devtools::install_github ("hydrocodes/hydRopUrban")
3
4 library(hydRopUrban)
5 data <- read.table(file.choose(), header=TRUE)
6 output <- "D:/2_Courses/R_Hidrologia/Tutorial_files/urban/output.txt"
7 dt <- 0.05 # time interval for plotting in hr (e.g. 0.05 or 3-min)
8 D <- 1
9 rational(data,dt)
10 rationalm(data,dt)
11 rationalu(data,dt)
12
13 caquot(data, a=5, b=-0.6)
14 caquotp(data, a=5, b=-0.6)
15
16
```

```
** testing if installed package can be loaded from final location
*** arch - i386
*** arch - x64
** testing if installed package keeps a record of temporary installation path
* DONE (hydRopUrban)
> library(hydRopUrban)
> data <- read.table(file.choose(), header=TRUE)
> output <- "D:/2_Courses/R_Hidrologia/Tutorial_files/urban/output.txt"
> dt <- 0.05 # time interval for plotting in hr (e.g. 0.05 or 3-min)
> rationalu(data,dt)
[1] 0.01948328
[1] 0.02329645
[1] "Qpeak = 0.041258 m3/s, Volume = 83.839531 m3"
```

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
```

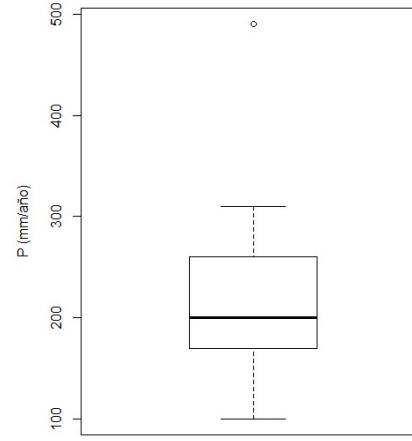
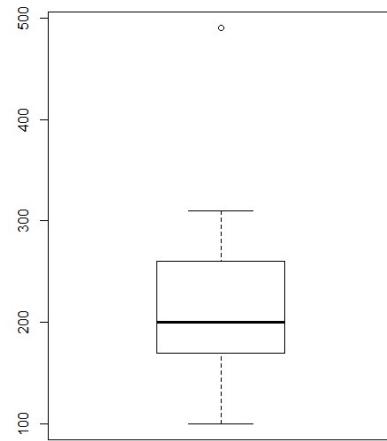
Ejercicio A.2

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] 231.375
> median(rain)
[1] 200.1
> sd(rain)
[1] 120.0709
> var(rain)
[1] 14417.03
> summary(rain)
   Min. 1st Qu. Median  Mean 3rd Qu.  Max.
100.5   180.0  200.1  231.4  235.2  490.0
> boxplot(rain)
> boxplot(rain, ylab="P (mm/año)") #Grafico de cajas y etiqueta en eje vertical
```



Ejercicio A.3

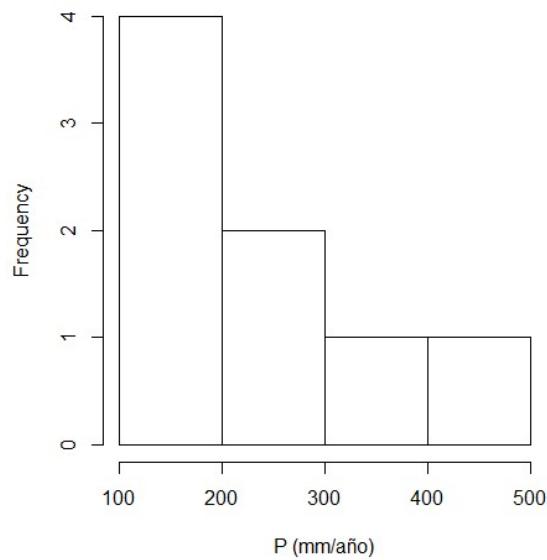
En la consola de resultados, efectuar las siguientes operaciones:

- 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.

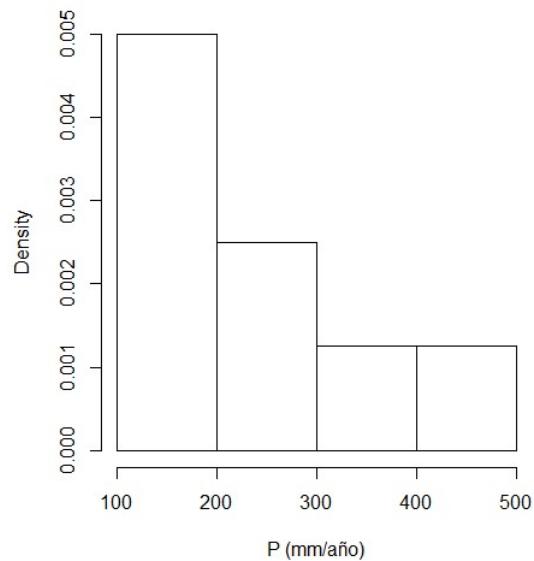
Respuestas A.3

```
> 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))
```

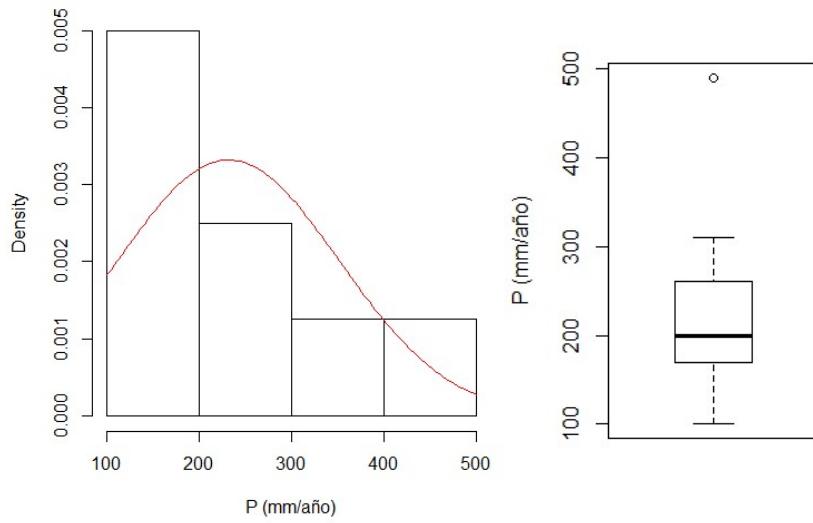
Precipitacion anual



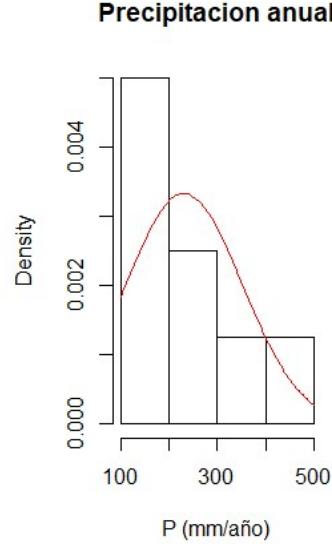
Precipitacion anual



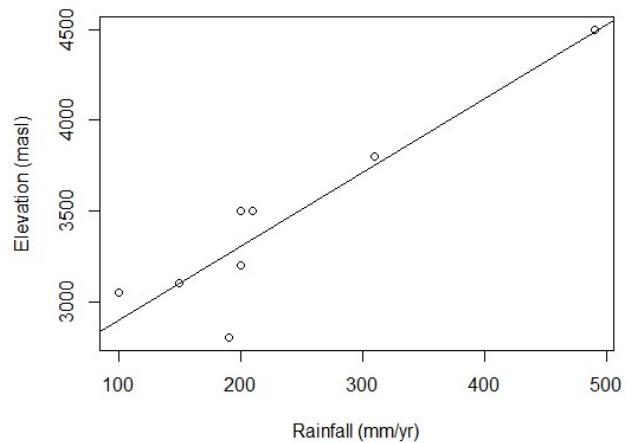
Precipitacion anual



Precipitacion anual



Elevation-Rainfall relationship



3. Análisis exploratorio y tratamiento 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, ...)
```

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 = "")

Leyendo archivos, bases de datos

Configurando Series de tiempo

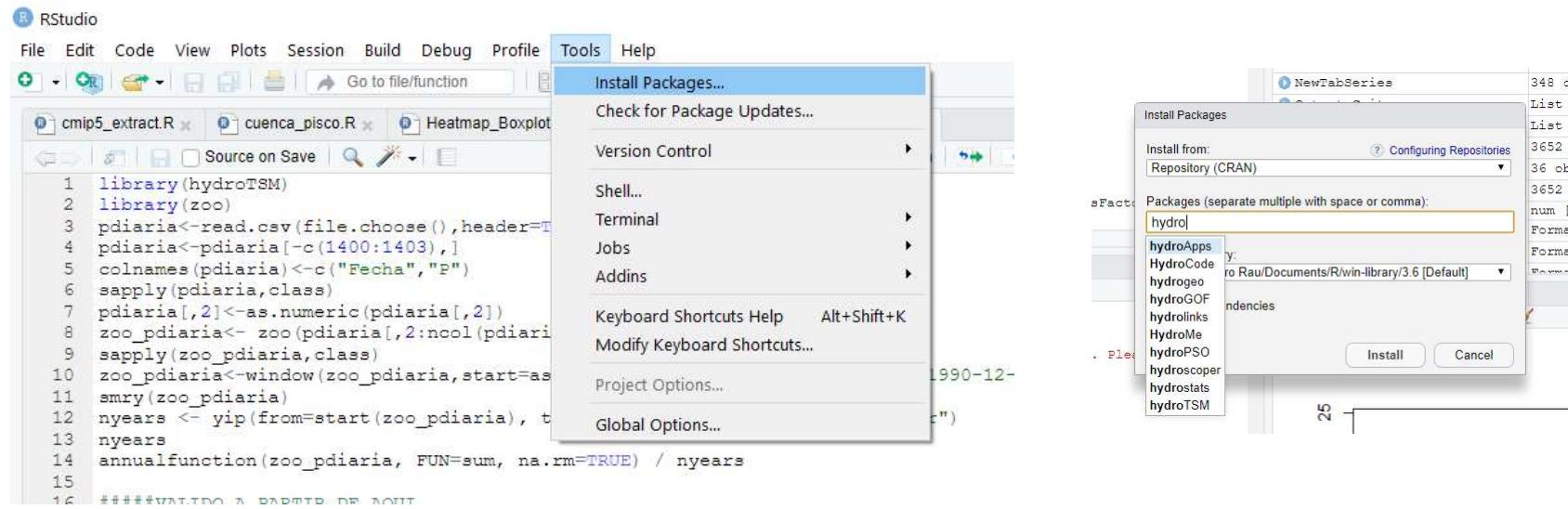
```
ts(data = NA, start = 1, end = numeric(0), frequency = 1, deltat = 1, ts.eps =  
getOption("ts.eps"), class, names)
```

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, ...

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://github.com/hydrocodes/Series_hidro_R/blob/master/Tutorial_files/pmensual.txt
- 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 valor1 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

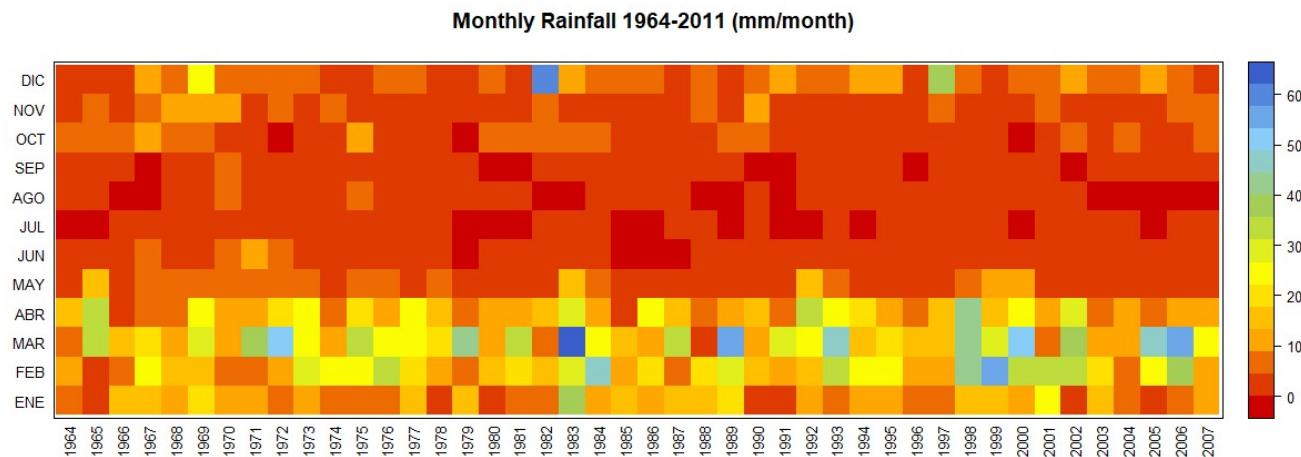
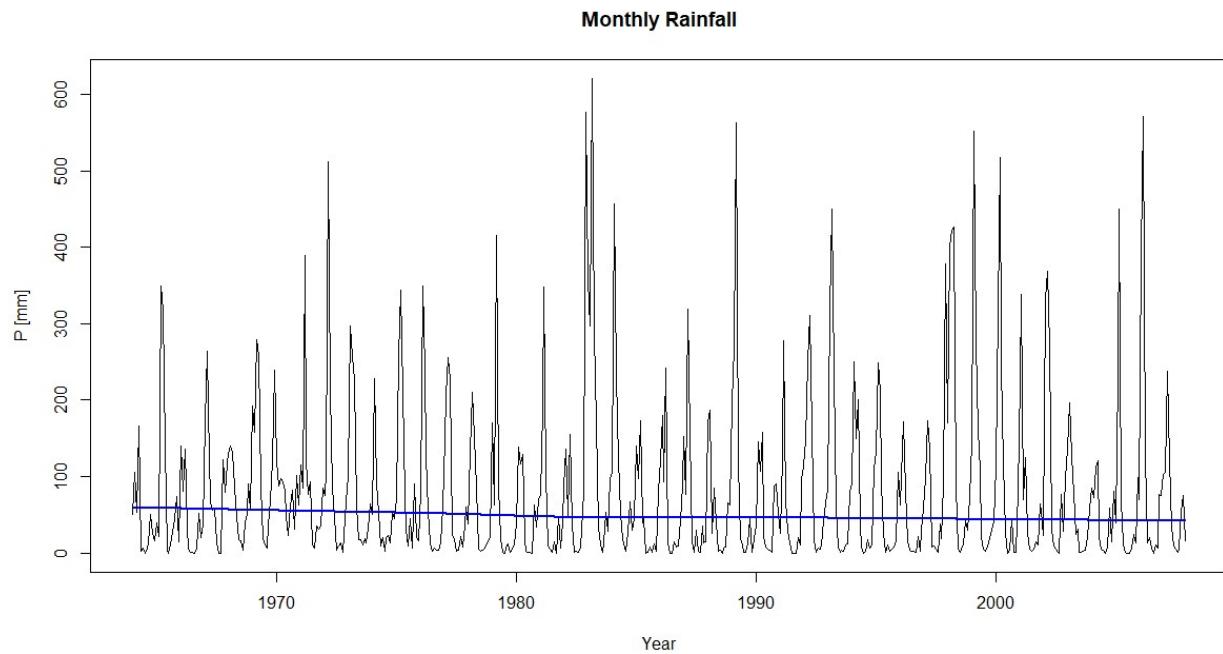
```
library("lattice") #llamar librería lattice  
library("hydroTSM") #llamar librería TSM  
pmensual<-read.table(file.choose(), header = F) #lectura de archivo, se abrirá una ventana para seleccionar el archivo, cual fuese su ruta, header "F" por no contener datos en la cabecera solo contiene etiquetas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

#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)")
```



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

Uso de las librerías: xts, lattice, ggplot2

- Instalar librerías TOOLS - INSTALL PACKAGES: *easypackages*, *xts*, *lattice*, *ggplot2*
- Descargar archivo "p_diarias.csv"
https://github.com/hydrocodes/Series_hidro_R/blob/master/Tutorial_files/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
- 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

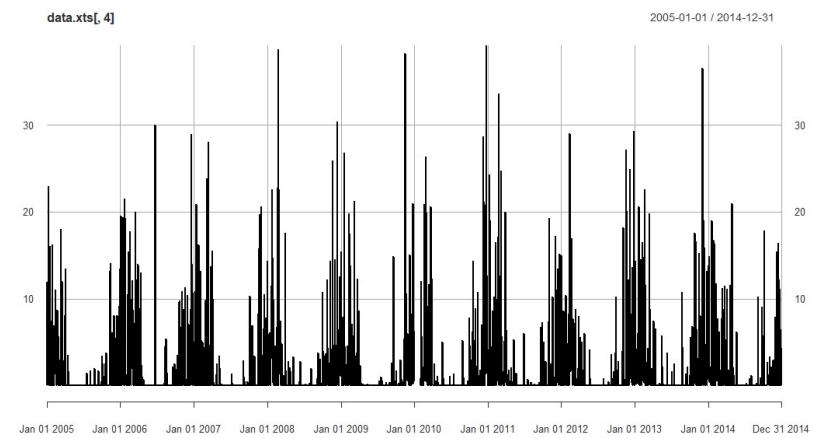
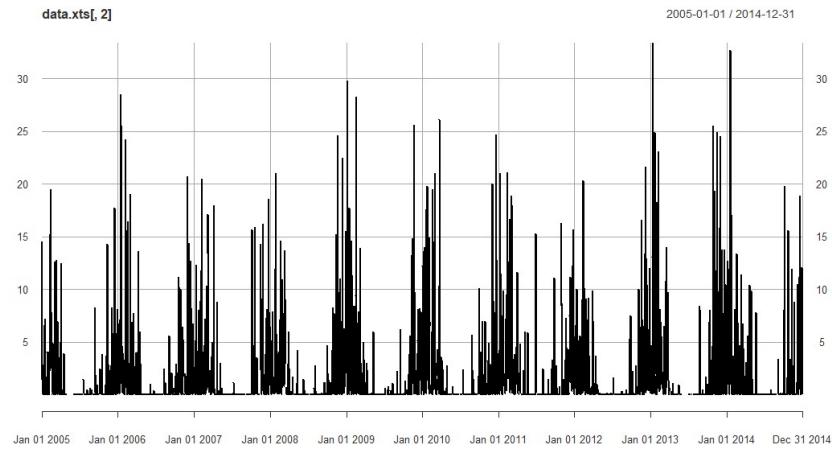
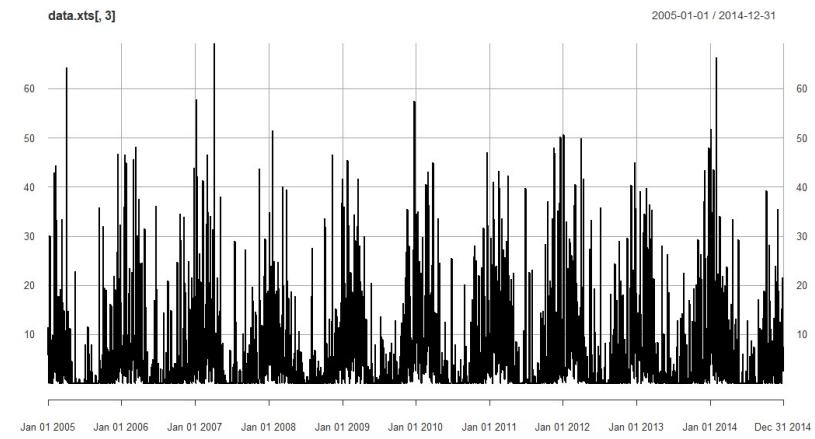
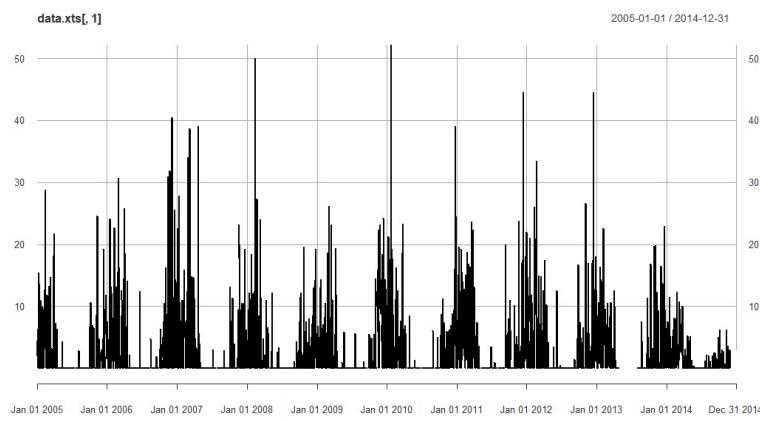
```
library(easypackages)
library(xts)
library(lattice)
library(ggplot2)
pdiaria <- read.csv(file.choose(),header=TRUE, check.names = F, stringsAsFactors = F)
str(pdiaria) #para ver la estructura del objeto
idx <- as.Date(pdiaria[,1]) #formato fecha a la 1era columna
data.matrix <- pdiaria[,-1] #formato 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))

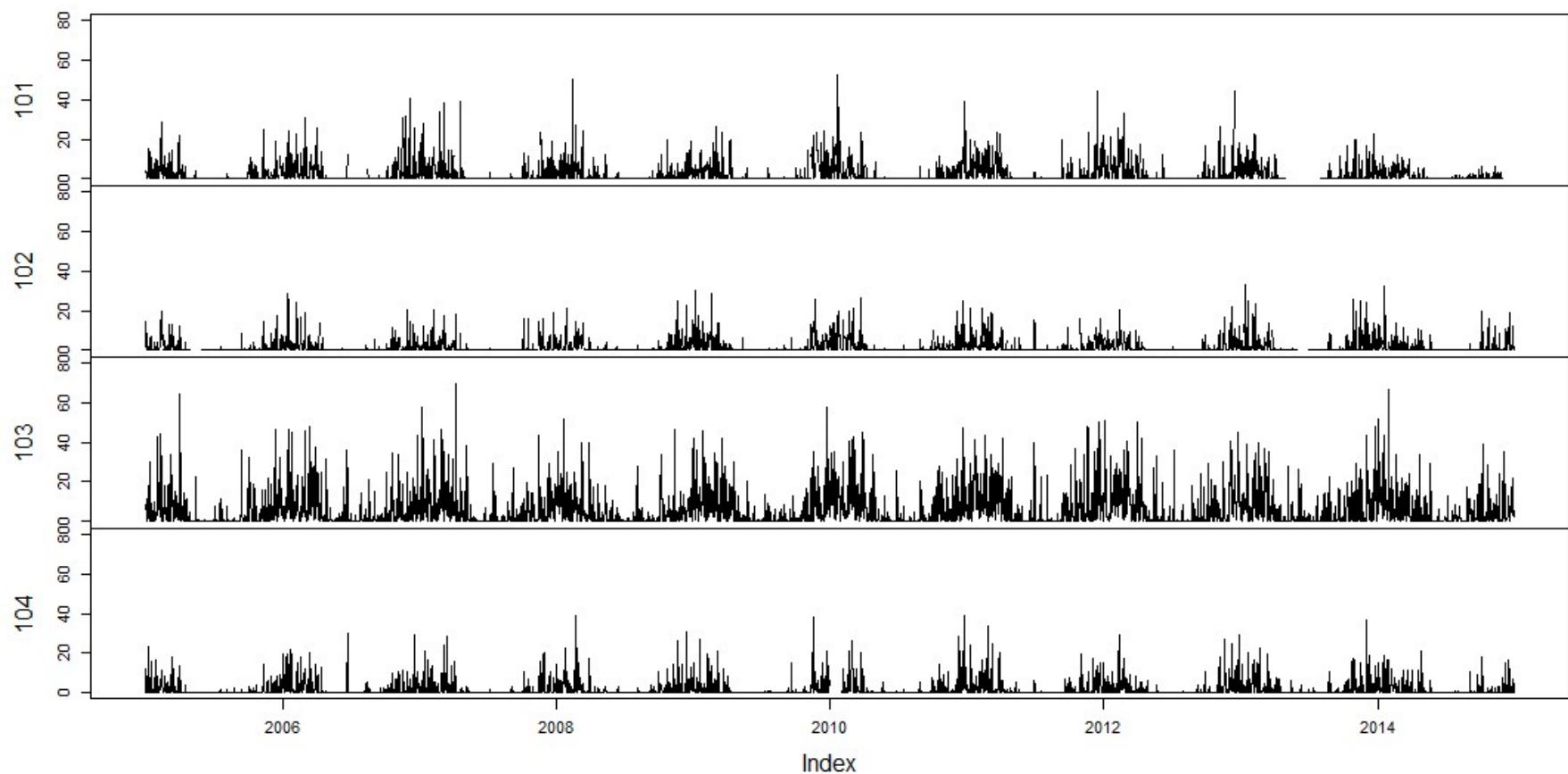
autoplot(data.xts[,1:2]) +theme_bw() +xlab('Fecha') +ylab('Precipitación [mm/dia]')
boxplot(coredata(data.xts))
hist(coredata(data.xts[,1]), freq = T) # cantidad de datos por clase
histogram(coredata(data.xts[,1])) # porcentaje de datos por clase

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

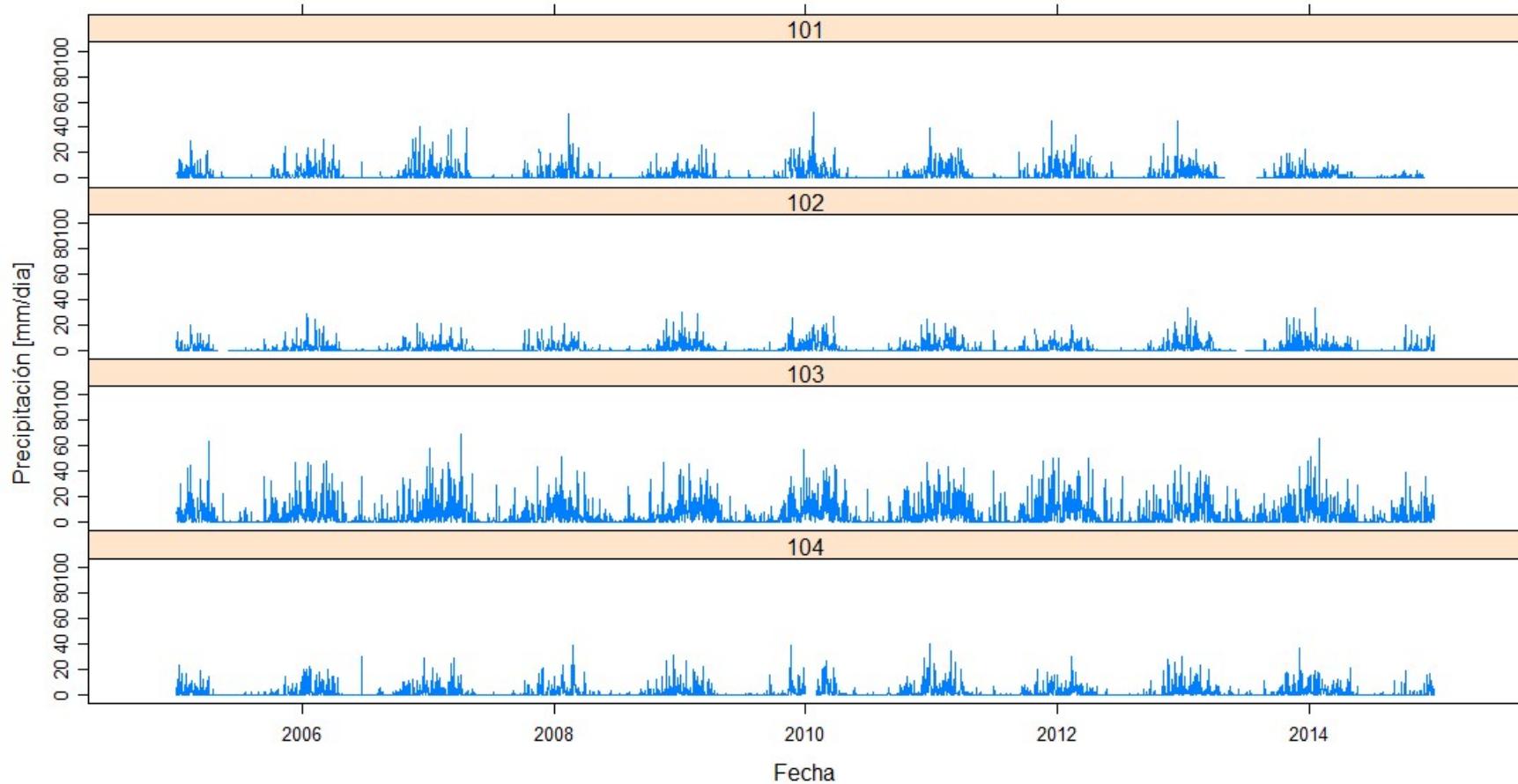


Ploteo simple

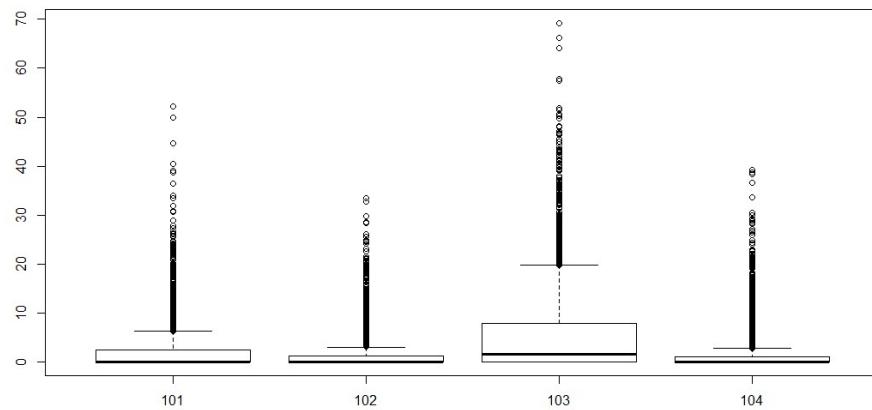
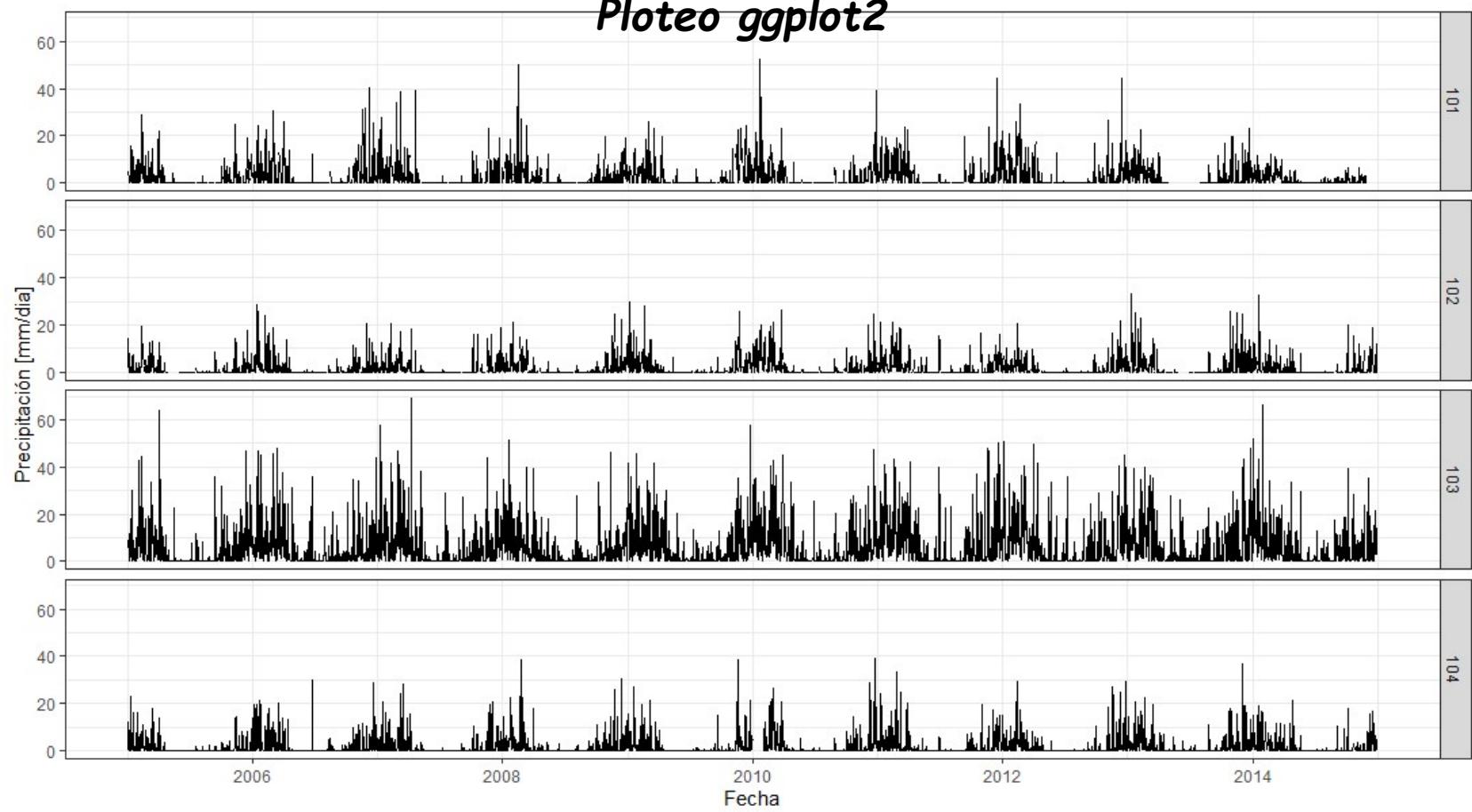
Series de tiempo de precipitación

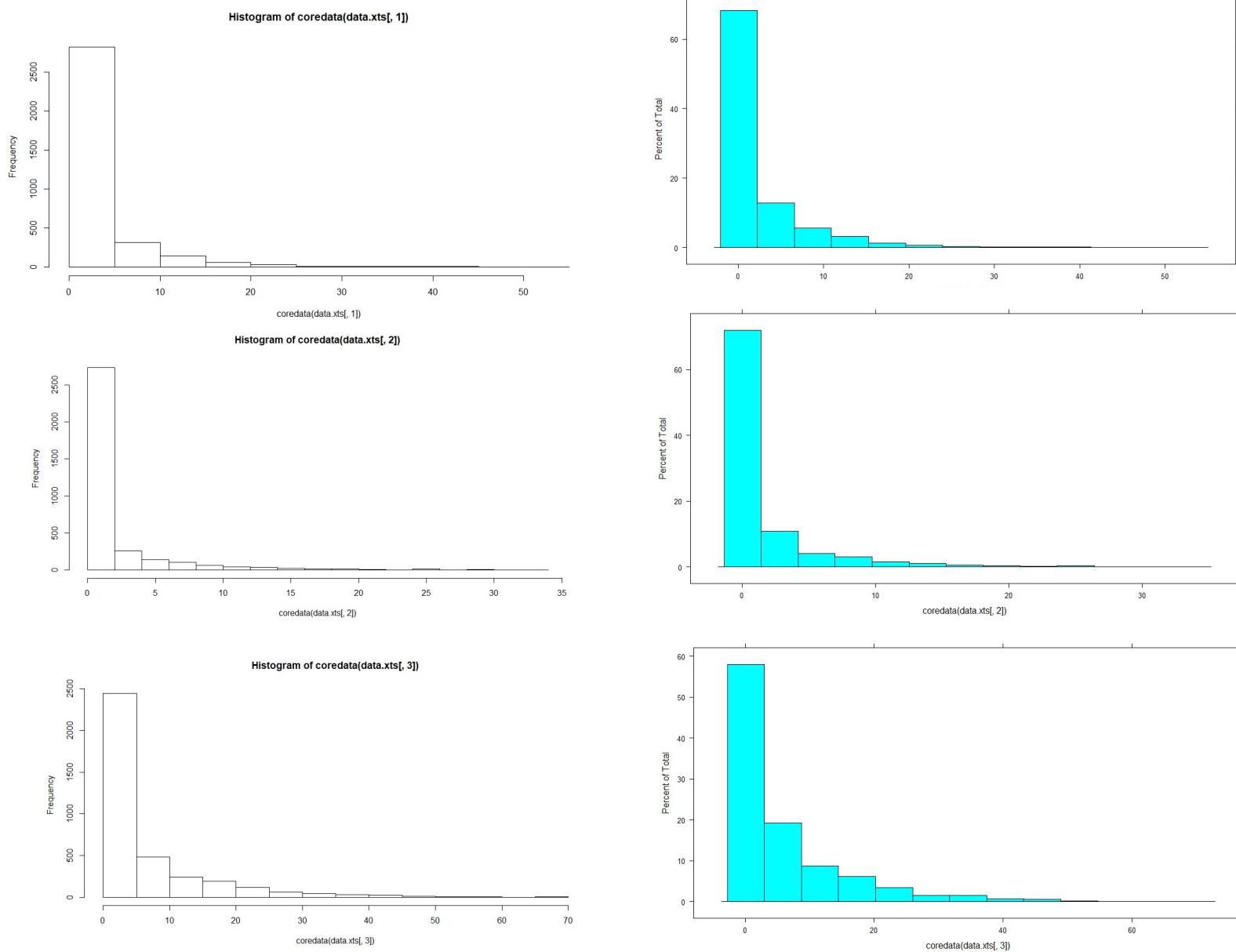


Ploteo lattice



Ploteo ggplot2





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 linea 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 estacion
- 4) Plotear un boxplot estacional de la estacion 103

Respuestas D

```
library(easypackages)
library(xts)
library(lattice)
library(ggplot2)
pdaria <- read.csv(file.choose(), header=TRUE, check.names = F, stringsAsFactors = F)
str(pdaria)
idx <- as.Date(pdaria[,1])
data.matrix <- pdaria[,-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))

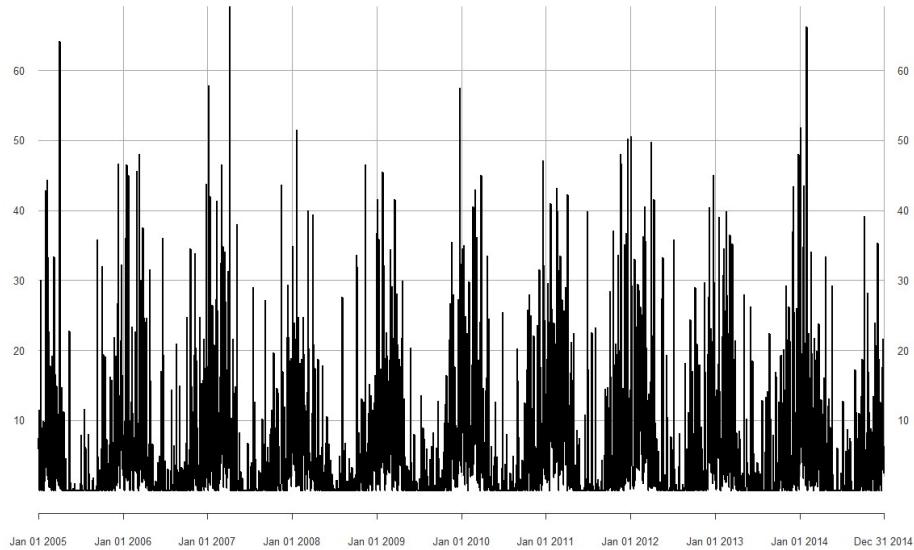
#Convirtiendo a anuales la estacion 103
data.anual <- apply.yearly(data.xts[,3], FUN = sum)
barplot(data.anual)

#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]"))
```

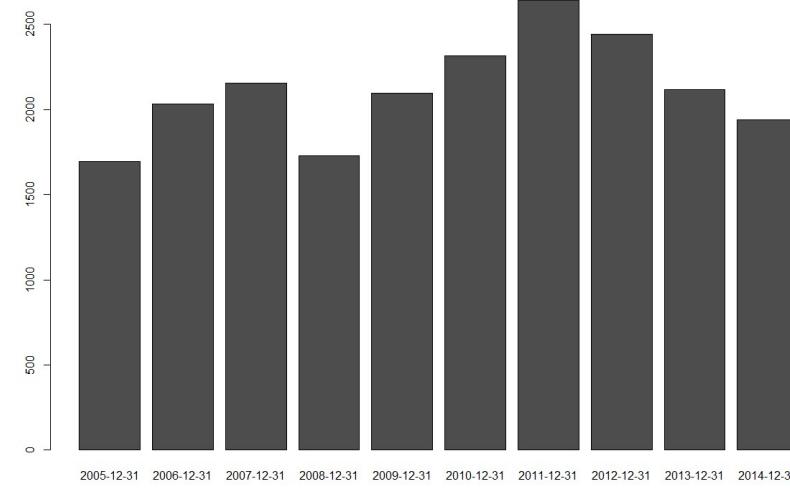
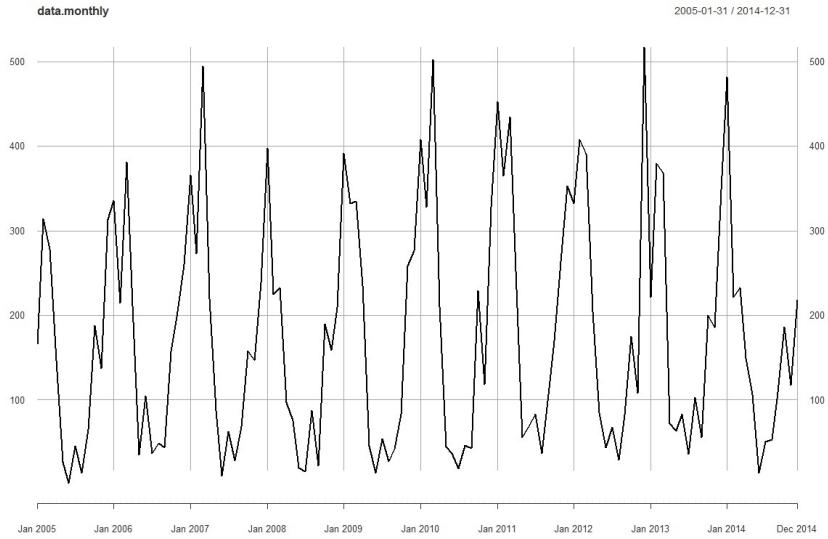
`data.xts[, 3]`

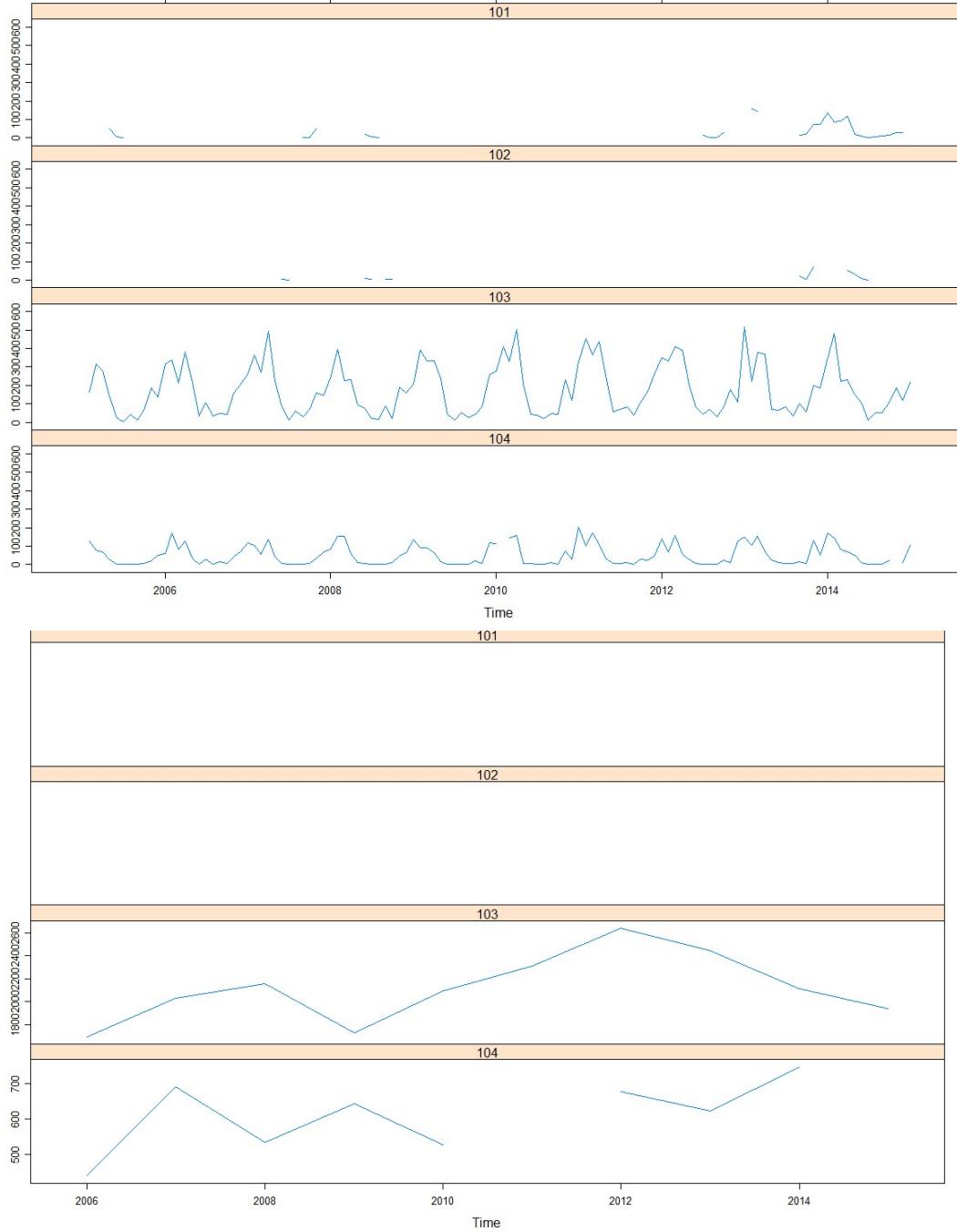
2005-01-01 / 2014-12-31

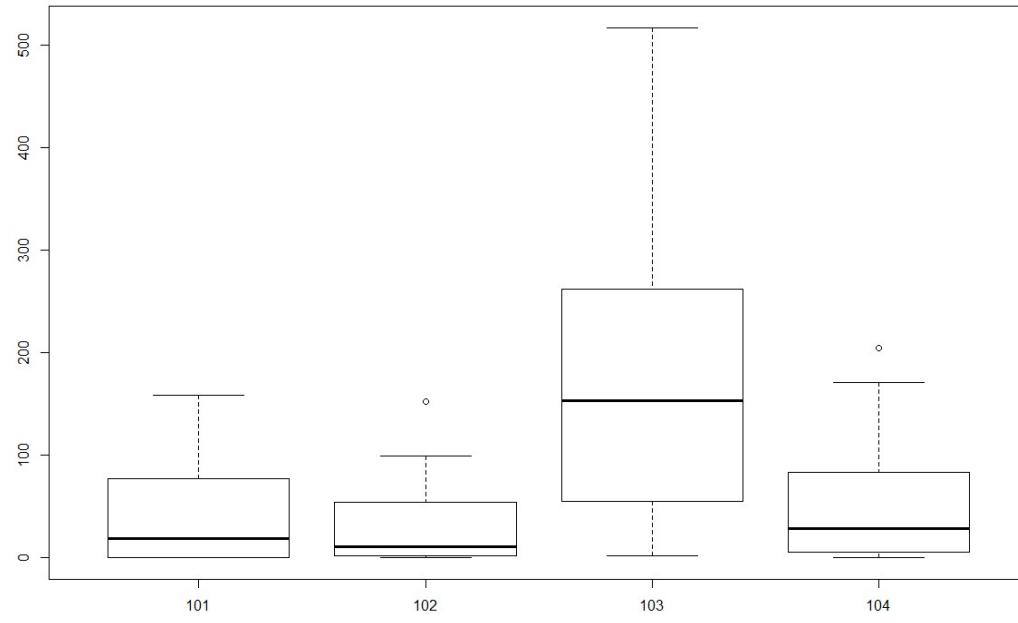


`data.monthly`

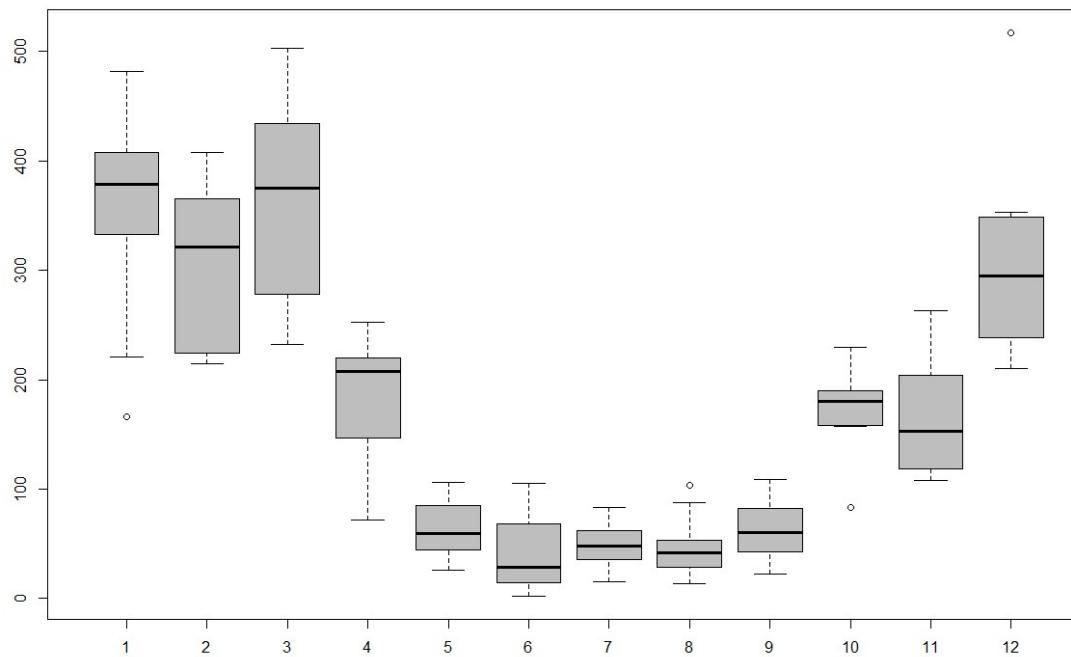
2005-01-31 / 2014-12-31







103
Prec mensual [mm]



E. Extracción de datos netcdf para una cuenca hidrográfica

-Instalar librerías TOOLS - INSTALL PACKAGES: **sp**, **rgdal**, **raster**, **lattice**, **latticeExtra**, **ncdf4**

-Revisión de archivos: Cuenca del río Vilcanota (*cuenca_vilcanota.shp* y archivos SIG anexos).

<https://drive.google.com/file/d/1xwFSrFcb5anOoAhf4Vz40Mqwd7IRXfzZ/view?usp=sharing>

-Revisión del producto grillado de precipitación mensual PISCO (Peruvian Interpolate data of the SENAMHI's Climatological and hydrological Observations, SENAMHI, Aybar et al. 2020) desde 1981 al 2016 (PISCOV3-MONTHLY.nc)

<https://drive.google.com/file/d/1efrg112gyFgGd24alNa3VNKr5bnQWjL4/view?usp=sharing>

<http://iridl.ldeo.columbia.edu/SOURCES/.SENAMHI/.HSR/.PISCO/index.html?Set-Language=es>

Ejercicio E

Extraer la serie de tiempo de precipitación mensual agregado para toda la cuenca del río Vilcanota, desde el producto PISCO para el periodo 1981 al 2016. Trabajar en el sistema de coordenadas geográficas WGS84.

Respuestas E

```
library(sp)
library(rgdal)
library(Rcpp) # for R 4.x versions, try also package "terra"
library(raster)
library(ncdf4)

# Definiendo un directorio de trabajo donde se encuentren los archivos input descargados
setwd("D:/1_Projects/002_Glaciares/02_RAHU/gis")

# Leyendo el polígono vilcanota.shp (layer: vilcanota) desde la carpeta "files" con data source name (dsn)
cuenca.shape <- readOGR(dsn="vilcanota", layer="cuenca_vilcanota")

# Conociendo el tipo de objeto después de la importación
class(cuenca.shape)

# Visualizando la cuenca con color cyan, se aprecia que se encuentra en coordenadas geograficas
plot(cuenca.shape, axes=T, col=c("cyan"))

# Conociendo el contenido del polígono importado
head(cuenca.shape@data)

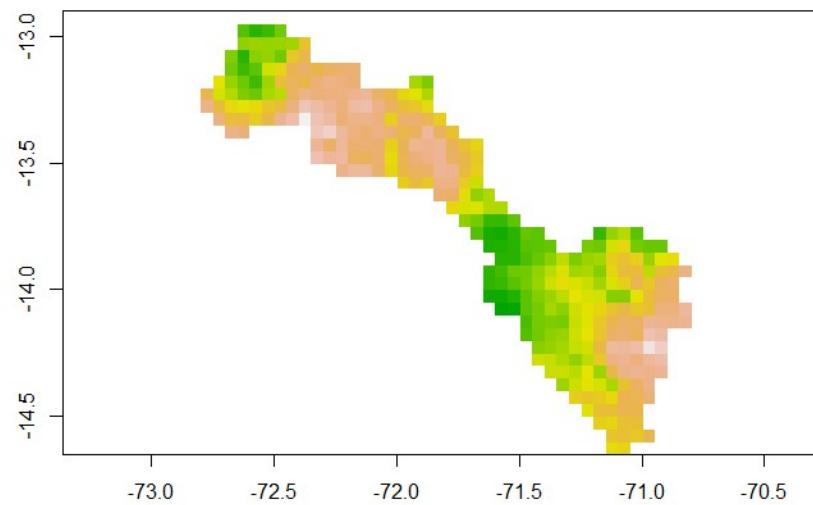
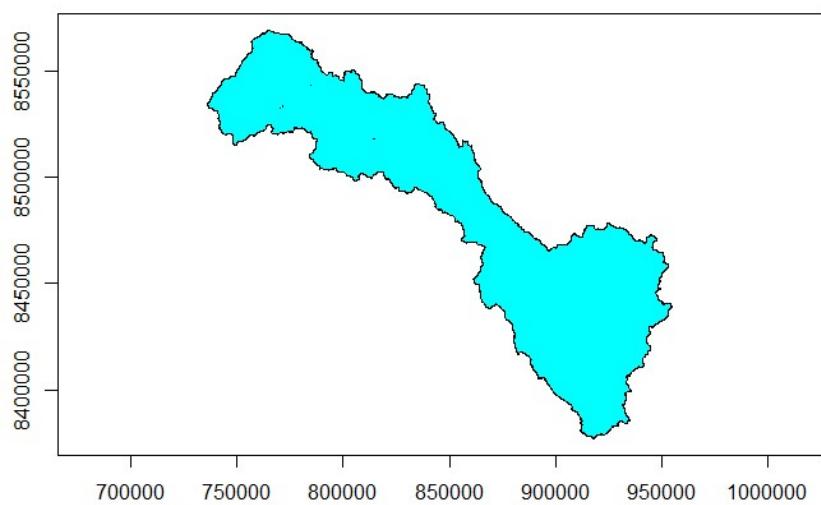
# Transformando a UTM zona 18, WGS84 y visualizando la conversión
cuenca.utm <- spTransform(cuenca.shape, CRS("+proj=utm +zone=18 +ellps=WGS84 +south
+datum=WGS84 +units=m +no_defs +ellps=WGS84 +towgs84=0,0,0"))
plot(cuenca.utm, axes=T, asp=1)

# Reproyectando a Geograficas WGS84 y visualizando la reconversión
cuenca.wgs <- spTransform(cuenca.utm, CRS("+proj=longlat +ellps=WGS84"))
plot(cuenca.wgs, axes=T, asp=1)

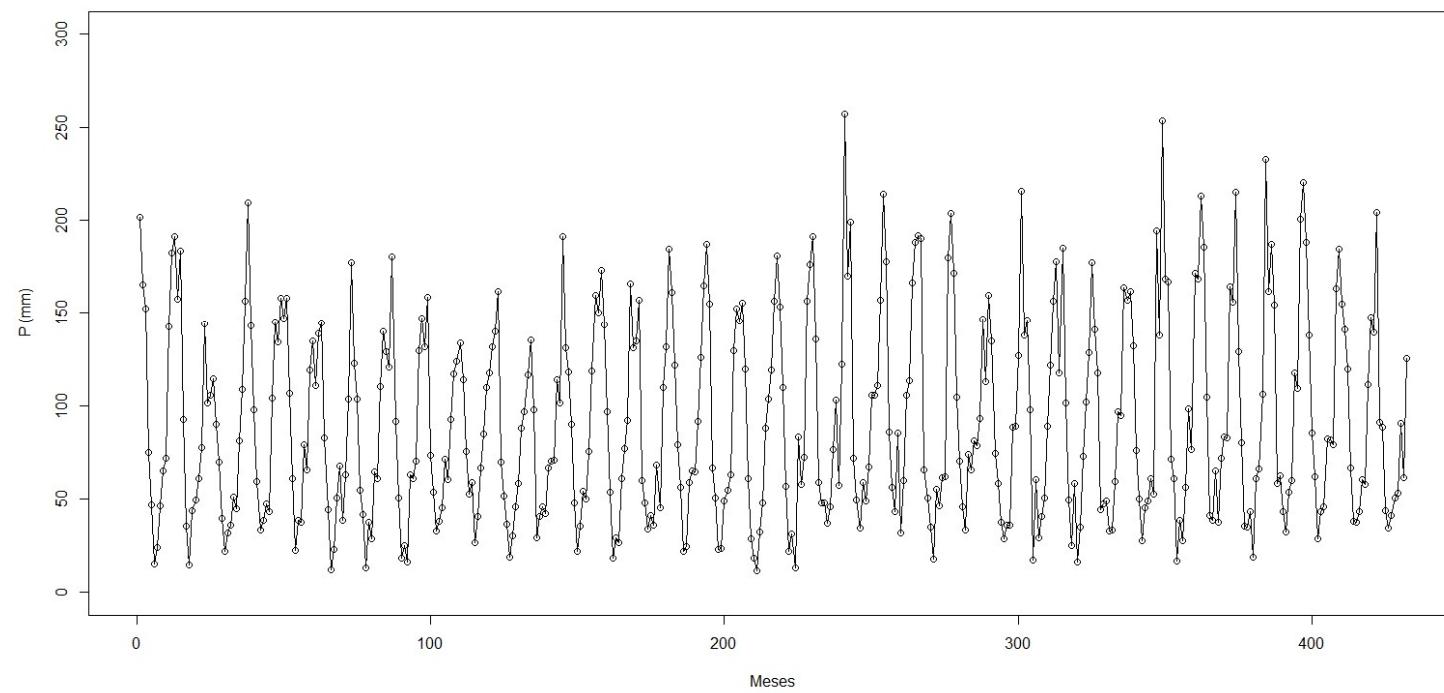
# Extrayendo datos y visualizando el producto ráster PISCO
r <- stack("PISCOV3-MONTHLY.nc") # El archivo *.nc debe estar en la ruta directa setwd
```

```
# Visualizando la precipitación espacial del primer mes (Enero) de 1981
plot(r[[1]])
# Ploteando la cuenca del río Vilcanota dentro del mapa de precipitaciones
plot(cuenca.wgs, add=T)
# Delimitando el área de estudio al cuadrante que ocupa el río Vilcanota
r.basin <- crop(r, cuenca.wgs, snap="out")
# Ploteando enero de 1981 en el cuadrante que ocupa el río Vilcanota
plot(r.basin[[1]])
# Delimitando el área de estudio a la cuenca del río Vilcanota
r.basin <- mask(r.basin, cuenca.wgs)
# Ploteando los meses de enero a diciembre de 1981
plot(r.basin[[1:12]])
# Extrayendo los datos y promediando todas las grillas de la cuenca del Vilcanota
pp.cuenca.mensual <- extract(r.basin, cuenca.wgs, fun=mean)
colnames(pp.cuenca.mensual) <- 1:ncol(pp.cuenca.mensual)
# Visualizando los 431 datos promediados
View(pp.cuenca.mensual)
range(pp.cuenca.mensual)

# Ploteando la serie de los 431 valores de precipitación mensual promedio areal
plot(pp.cuenca.mensual[1,], type="o", col="1", ylim=c(0,300), ylab="P (mm)", xlab = "Meses",
main="Precipitación promedio areal - Vilcanota (mm)")
```



Precipitacion promedio areal - Vilcanota (mm)



Referencias

Aybar C, Fernández C, Huerta A, Lavado W, Vega F, Felipe-Obando O. 2020. Construction of a high-resolution gridded rainfall dataset for Peru from 1981 to the present day. *Hydrol. Sci. J.* 65, 770-785.

Ihaka R. & Gentleman R. 1996. R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5: 299-314.

Paradis E. 2002. R for beginners. Institut de Sciences de l'évolution. Université de Montpellier. France

Rau P, Bourrel L, Labat D, Melo P, Dewitte B, Frappart F, Lavado W, Felipe O, 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



Visita los enlace de nuestro Proyecto!

RAHU: Glaciares y Seguridad Hídrica

<https://sites.google.com/utec.edu.pe/rahu-project/rahu>

<https://www.facebook.com/RAHU.seguridad.hidrica>