Estadística con R

José Ulises Jiménez S. y José Rogelio Fábrega D.

2021-07-20

Introducción

Este material complementario se preparó con el propósito de ser una guía básica de introducción a la computación estadística y al análisis de datos con el sistema `R` para el curso de estadística que se dicta a los estudiantes de pregrado de la Universidad Tecnológica de Panamá. Le llamamos material complementario porque sobre `R` existen muchísimos libros, folletos y guías; además de *blogs* y tutoriales, que están disponibles en la Internet de manera gratuita.

`R` es un lenguaje y una interfaz para: el manejo y la manipulación de datos, el análisis estadístico, la programación y la simulación científica, la creación de gráficos altamente sofisticados y trabajar con otros programas (*WinBUGS*, *Mezquite*, *GIS*, etc.).

Los objetivos principales de un curso de Estadística con `R` son: primero, mostrar a los estudiantes cómo se utiliza el razonamiento estadístico en la investigación; segundo, permitir a los estudiantes realizar con confianza análisis estadísticos sencillos e interpretar los resultados; y último, sensibilizar a los estudiantes sobre cuestiones estadísticas básicas en cuanto a la calidad de los datos como: la aleatorización y el que las replicas sean independientes.

En la actualidad, los análisis de los datos en las investigaciones se realizan con la ayuda de una computadora. Los cálculos no se hacen a lápiz y papel, ni los gráficos se dibujan a mano. De allí que las herramientas computacionales son hoy en día un complemento imprescindible en el aprendizaje de la estadística. Esta experiencia específicamente preparará al estudiante para hacer un uso apropiado del lenguaje y entorno `R` para el aprendizaje de la estadística. Aprenderá a darle las instrucciones al programa `R` (secuencias de código) y a interpretar las salidas o resultados. La enseñanza de la estadística además de brindar una base matemática clara, debe enfatizarse en:

La estadística computacional es un componente clave de la ciencia de los datos, definida como la habilidad de usar datos para responder preguntas y comunicar resultados.

1. La correcta selección del diseño experimental, la calidad de los datos, los procedimientos y pruebas.
2. La interpretación de las salidas y resultados.
3. La utilización de programas estadísticos gratuitos y confiables como `R`, que permiten la reproducibilidad (se proporcionan los datos y los códigos) y replicabilidad (se proporcionan solo los códigos) de las investigaciones.

Versión R `4.0.5`.

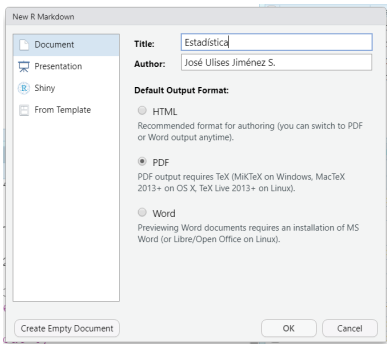
Los apuntes del curso de estadística se escribieron para cubrir las necesidades básicas de competencias en estadística de los estudiantes de pregrado y aprovecha la versión 4.0.5 o posterior de `R`, en un sistema operativo *Windows* para el análisis de datos.

Este documento y los códigos fueron escritos usando `R Markdown`,

`knitr` y el estilo de folleto `Tufte` del paquete del mismo nombre, que proporciona plantillas para crear folletos según el estilo de Edward R. Tufte y Richard Feynman (Xie & Allaire, 2020). `R Markdown` proporciona un marco flexible para mezclar texto y código `R` para la generación automática de documentos. Los documentos `R Markdown` se pueden convertir en una variedad de formatos como: `HTML`, `PDF`, `MS Word` y `Beamer`.

El paquete `knitr` proporciona una herramienta para la generación de informes dinámicos en `R` utilizando técnicas de programación alfabetizada, lo cual lo convierte en una herramienta alternativa a `Sweave` basada en un diseño diferente con más funciones (Xie, 2021, 2015, 2014).

Dentro del documento `R Markdown` se pueden incluir, ya sea el caso, códigos de `HTML` o de LaTeX (debes tener instalado *MiKTeX* en tu computador para convertir estos documentos a `PDF`). LaTeX es un sistema de preparación de documentos para composición de texto de alta calidad.



Los documentos `R Markdown` se pueden convertir en una variedad de formatos.

Objetivos específicos

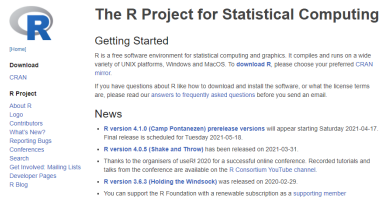
Los objetivos específicos de las clases prácticas del curso de estadística con `R` son:

1. Introducir los conceptos básicos de `R`.
2. Familiarizar a los estudiantes con el uso de `R` y `RStudio` para el análisis estadístico.
3. Mostrar como ingresar los datos en un formato que le guste a `R`.
4. Presentar el uso de paquetes y funciones útiles para aprender estadística.
5. Desarrollar ejemplos concretos de análisis estadísticos y realizar prácticas.
6. Enseñar a preparar documentos básicos con `Rmarkdown` (Allaire et al., 2021; Xie et al., 2018, 2020).
7. Indicar como obtener ayuda para seguir avanzando por su cuenta en la estadística y el uso de `R`.

Logo de `R`.

Acerca de R

`R` es un conjunto integrado de funciones de software para manipulación de datos, cálculo y visualización gráfica. Este documento trata sobre funciones escritas en `R`, un lenguaje y entorno para la informática estadística (R Core Team, 2021). Visite la página web `http://www.R-project.org` para conocer más sobre `R`.



`http://www.R-project.org`

¿Qué es R?

1. `R` es un programa de última generación para realizar análisis de datos, siendo también un lenguaje de programación, lo cual lo hace muy versátil.
2. Como lenguaje de programación, `R` es un dialecto de un lenguaje de programación denominado `S`.
3. Dentro de los lenguajes de programación se puede clasificar como un lenguaje orientado a objetos de tipo interpretado. Lo que lo hace flexible, potente y posee un tiempo de aprendizaje corto.
4. Actualmente se encuentran disponibles 17896 paquetes desarrollados en `R` en el *Comprehensive R Archive Network* (CRAN, por sus siglas en inglés), que cubren multitud de campos desde aplicaciones Bayesianas, financieras, *wavelets*, análisis de datos espaciales, etc.

Actualmente, el repositorio de paquetes CRAN (La Red Integral de Archivos `R`) presenta `17896` paquetes disponibles. CRAN es una colección de sitios que contienen material idéntico, que consta de las distribuciones de `R`, las extensiones aportadas, la documentación para `R` y los archivos binarios.

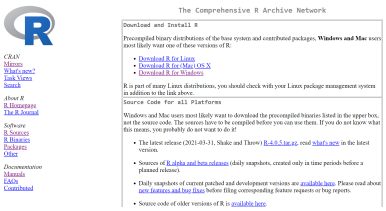
Notas históricas

1. 1991: creado en Nueva Zelanda por Ross Ihaka y Robert Gentleman.
2. 1993: primer anuncio público de `R`.
3. 1995: Martin Machler convence a Ross y Robert para hacer de `R` un *sofware* libre bajo la licencia GNU.
4. 1997: se crea el núcleo de desarrollo del código fuente de `R`, que pasa a controlar todo lo relativo al desarrollo del código fuente.
5. 2000: aparece la versión 1.0.0 de `R`.
6. En este momento, la versión más actualizada lleva el número 4.1.0 (2021-05-18).

Características de R

1. `R` proporciona muchísimas herramientas estadísticas para el análisis de datos: modelos lineales y no lineales para regresión, pruebas estadísticas, análisis de series temporales, algoritmos de clasificación y agrupamiento, gráficas, etc.
2. Como es un lenguaje de programación, permite que los usuarios lo extiendan definiendo sus propias funciones. Gran parte de las funciones de `R` están escritas en el mismo `R`, aunque para algoritmos computacionalmente exigentes es posible desarrollar bibliotecas en `C`, `C++` o `Fortran` que se cargan dinámicamente. La sintaxis es relativamente simple.
3. `R` también puede usarse como herramienta de cálculo numérico, donde puede ser tan eficaz como otras herramientas específicas tales como `GNU Octave` y su equivalente comercial, `MATLAB`.
4. Es multiplataforma, se puede ejecutar en casi todas los sistemas operativos (*Linux*, *Windows*, *MacOS*).

5. Existe una comunidad muy extendida que lo mantiene en permanente actualización.
6. Está dividido en paquetes modulares que responden a necesidades específicas.
7. Posee excelente capacidades gráficas.
8. Posee un excelente paquete, `knit`, que permite desarrollar presentaciones interactivas que evalúan el código `R` en el momento.
9. Es libre.

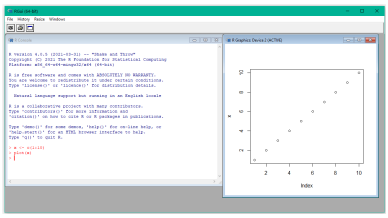


Comprehensive R Archive Network.

Diseño del sistema R

El sistema `R` está dividido en dos partes:

1. El sistema `base`, que puede ser descargado desde la red de servidores CRAN (*Comprehensive R Archive Network*) `http://cran.r-project.org`, contiene las funciones fundamentales y todo lo requerido para que R funcione. Es sistema base contiene los paquetes: `utils`, `stats`, `datasets`, `graphics`, etc. También, contiene paquetes recomendados: `boot`, `class`, `cluster`, `codetools`, etc.
2. Todo el resto de paquetes pueden ser descargados desde CRAN. Adicional a los paquetes en CRAN existe una innumerable cantidad de paquetes en sitios personales y que son de libre uso.

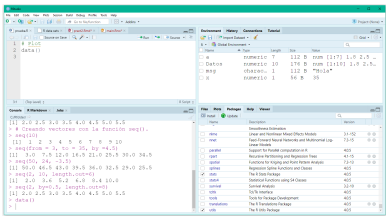


RGUI.

Uso de la consola y las GUI

Hay dos maneras de interaccionar con `R`:

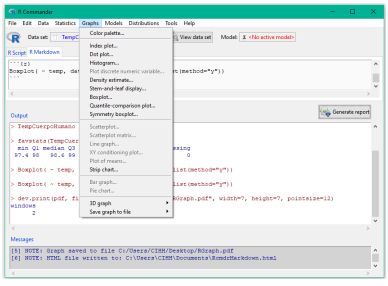
1. Por la consola, RGUI.
2. Mediante una interfaz gráfica de usuario (GUI, por sus siglas en inglés) que hace de puente entre el programa y el usuario.
   a. RStudio
   b. RCommander
   c. RKWard
   d. etc.



GUI de RStudio.

Es conveniente instalar una interfaz gráfica como Rstudio. Puede encontrar los archivos de instalación en la página web `http://www.rstudio.com`. También, se puede usar `RStudio Cloud` que es una aplicación en un servidor accesible a los usuarios vía Internet donde solo tienes que registrarte.

Con `RStudio Cloud` no hay que instalar nada en su computadora, de esta forma se evitan todos los potenciales problemas con particularidades de la computadora personal de cada estudiante. Para acceder a `RStudio Cloud` visita la página web `https://rstudio.cloud/`.

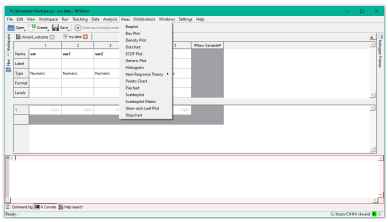


GUI de R Commander.

Fuentes para aprender R

Disponibles en CRAN `http://cran.r-project.org`.

1. *An Introduction to R* en `https://cran.r-project.org/doc/manuals/R-intro.html`.
2. *Writing R Extensions.*
3. *R Data Import/Export.*
4. *R Installation and Administration (mostly for building R from sources).*
5. *R Internals.*
6. *CRAN Task Views* en `http://cran.r-project.org/web/views/`.



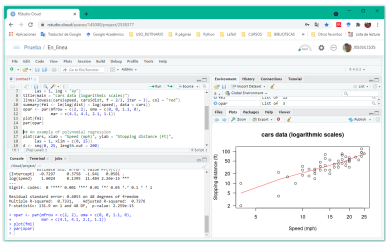
GUI de RKWard.

Otras fuentes

Artículo de *Wikipedia* y referencias citadas en `http://es.wikipedia.org/wiki/R_(lenguaje_de_programacion)`.

Instalación de R y RStudio

Primero debe instalar `R`, que es un programa básico con una interfaz simple, RGUI, que se instala de forma automática con el paquete `base` de `Windows`; segundo, debe instalar `RStudio`; tercero debe instalar `Rtools`; y por último, instale *MiKTeX*. En este apartado se muestra como encontrar los archivos ejecutables para la instalación de `R`, `RStudio`, `Rtools` y *MiKTeX*. Para instalar el sistema `R` y `Rtools`, se deben seguir las instrucciones de la página de CRAN `http://cran.r-project.org`.



RStudio Cloud.

Instale la versión correcta de `R`, `RStudio`, `Rtools` y *MiKTeX* para el sistema operativo de su computadora. Para la instalación de los programas `R`, `RStudio` y `Rtools`, simplemente acepte los valores predeterminados ofrecidos por los programas durante la instalación. Para instalar `RStudio` vaya a la página `http://rstudio.org/`.

Siga las instrucciones para instalar la versión correcta de `RStudio` en su computadora. En esta primera sesión práctica del curso de estadística se mostró paso a paso como instalar los programas: `R` que es el que realiza los cálculos, `RStudio` que es un entorno de trabajo, así que nos permitirá trabajar con `R` de forma cómoda y eficiente, `Rtools` que ayuda a terminar de compilar algunos paquetes para su instalación.

MiKTeX es necesario para tejer documentos pdf con el documento dinámico `R Markdown`. Para instalar *MiKTeX* vaya a la página `https://miktex.org/download`. Adicionalmente, instalaremos el programa `notepad++` para leer, analizar y editar los `script`. Para instalar `notepad++` vaya a la página `https://notepad-plus-plus.org/downloads/`.

Rstudio

El aspecto general de `RStudio` es su configuración con 4 paneles.

`RStudio` (RStudio Team, 2021) debe mostrar cuatro paneles. Puedes cambiar el tamaño de los paneles arrastrando en las divisiones. Arriba a la izquierda está el editor de códigos (escribes los `scripts` o guiones):

En el panel superior izquierdo puedes ingresar líneas de código en un archivo `script` antes de ejecutarlas.

1. Puedes escribir las secuencias de códigos antes de ejecutarlos.
2. Puedes escribir otros tipos de documentos donde están incrustados códigos y salidas (resultados) de `R` de manera automática.
3. `RStudio` permite que se abran muchos archivos de `script` y utiliza pestañas para ayudarlo a realizar un seguimiento de ellos.

Abajo a la izquierda está la `consola` y es donde se ejecutan los códigos.

El panel inferior izquierdo es la `consola` donde puedes escribir órdenes o secuencias de código y ver sus respuestas, como una calculadora. Además, si usa un archivo de `script` y ejecuta sus órdenes haciendo clic en `run` o apretando `ctrl+enter`, la salida de texto aparecerá en la consola.

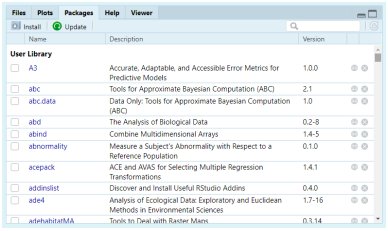
En el panel superior derecho, verá las variables y sus valores después de declararlas.

Arriba a la derecha hay un panel multifunciones:

1. Una de las ventanas es el entorno (*Environment*) que nos muestra los objetos o variables definidas con información de ellas.
2. El histórico nos muestra los códigos que hemos usado y nos permite re-ejecutarlas o enviarlas al `script`.

`R` es muy complejo, con miles de instrucciones y funciones con muchos parámetros y opciones, ni los que usamos R a menudo nos acordamos de todo, por lo que usamos mucho la ayuda.

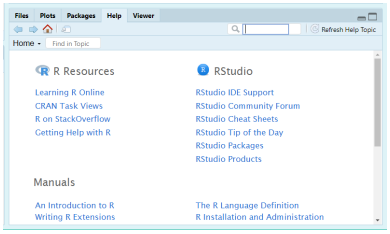
Finalmente, el panel inferior derecho (multifunciones) muestra información de ayuda y los diagramas que crea. Permite hacer muchas cosas como: encontrar archivos, visualizar gráficos, instalar y llamar paquetes, buscar ayuda y ver documentos.



Biblioteca de usuario.

Instalación de paquetes

1. Una vez instalado el sistema `base`, para instalar paquetes adicionales se utiliza la función: `install.packages("nombre del paquete")`.
2. Muchos de estos paquetes, además de poseer funciones poseen paquetes de datos.
3. Luego de instalar los paquetes, antes de usarlos debe llamarse en cada sesión de `R` y se utiliza la función: `library(nombre del paquete)`.
4. `RStudio` facilita la instalación de paquetes con las opciones del menú.
5. La funcionalidad de `R` se amplia con la instalación de los paquetes.



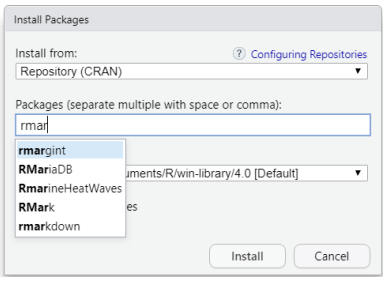
Accediendo a la ayuda.

```
install.packages("reshape", dep=TRUE)
library(reshape)
```

Conjunto de datos que acompañan las librerías de R

Además de las funciones asociadas en librerías específicas, R posee paquetes de datos. Si se escribe en la consola `data()` del paquete `utils`, se obtiene una lista de todos los conjuntos de datos que se encuentran en el paquete `datasets` y los paquetes llamados que tengan `data sets` (conjuntos de datos), principalmente `dataframes` que se encuentran dentro del entorno de trabajo.

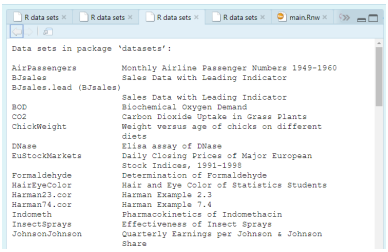
Para información sobre un `data set` en particular se utiliza el argumento `package` ajustado al nombre del paquete entre comillas, `data(package="nombre del paquete")`. Con `data(package = .packages(all.available = TRUE))`se listan los `data sets` de todos los paquetes disponibles.



Instalación de los paquetes.

```
data()
library(HistData)
data(package="HistData")
data(package = .packages(all.available = TRUE))
```

Podemos inspeccionar los objetos de datos creados con `head()`, `tail()`, `str()`, `dim()`, `class()` y `names()`.


Data sets en el paquete datasets.

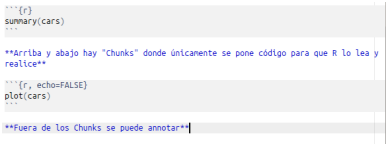
```
data("women")
bkmujer <- women
head(bkmujer)  # Las primeras seis observaciones.
```

```
##   height weight
## 1     58    115
## 2     59    117
## 3     60    120
## 4     61    123
## 5     62    126
## 6     63    129
```

```
tail(bkmujer)   # Las últimas seis observaciones.
```

```
##    height weight
## 10     67    142
## 11     68    146
## 12     69    150
## 13     70    154
## 14     71    159
## 15     72    164
```

```
str(bkmujer)  # Estructura de un objeto.
```


Los *chunks* permiten incrustar y ejecutar código R junto con el texto.

```
## 'data.frame':    15 obs. of  2 variables:
##  $ height: num  58 59 60 61 62 63 64 65 66 67 ...
##  $ weight: num  115 117 120 123 126 129 132 135 139 142 ...
```

```
class(bkmujer)  # Clase o tipo de objeto.
```

```
## [1] "data.frame"
```

```
names(bkmujer)  # los nombres de las columnas o variables.
```

```
## [1] "height" "weight"
```

La función `ls()` lista los objetos actuales. Con la función `rm()` puedes eliminarlos de Entono Global.

```
z <- c(5.6, 8.9, 7.5) # Creo un objeto z con tres valores
ls() # Muestra los objetos actuales
```

```
## [1] "bkmujer" "women"   "z"
```

```
rm(z) # borra el objeto
ls() # Muestra los objetos actuales
```

```
## [1] "bkmujer" "women"
```

Sobre R existen muchísimos libros, folletos y guías.

`HistData` es un paquete de datos que provee un conjunto de datos importantes en el desarrollo histórico de la Estadística. Si se escribe en la consola `sessionInfo()` se obtiene la versión de `R` y la lista de todos los paquetes (librería de funciones y de datos) que están incorporados a la sesión de `R` (`RStudio.Version()` para `RStudio`. Si se desea incorporar otro paquete de datos, se debe escribir `library(nombre del paquete)` en la consola, por ejemplo `library(HistData)`. Con `search()` se listan los paquetes que llamó con `library` y los básicos. Si desea saber la dirección donde se guardan los paquetes use `.libPaths()`.

```
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
```

```
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] tufte_0.9
##
## loaded via a namespace (and not attached):
##  [1] compiler_4.0.5    magrittr_2.0.1    tools_4.0.5       htmltools_0.5.1.1
##  [5] yaml_2.2.1        stringi_1.6.1     rmarkdown_2.8     knitr_1.33
##  [9] stringr_1.4.0     xfun_0.22         digest_0.6.27     rlang_0.4.11
## [13] evaluate_0.14
```

```
search()
```

```
##  [1] ".GlobalEnv"        "package:tufte"     "package:stats"
##  [4] "package:graphics"  "package:grDevices" "package:utils"
##  [7] "package:datasets"  "package:methods"   "Autoloads"
## [10] "package:base"
```

```
.libPaths()
```

```
## [1] "C:/Users/CIHH/Documents/R/win-library/4.0"
## [2] "C:/Program Files/R/R-4.0.5/library"
```

Encontrar las librerias que necesitamos

En el repositorio de CRAN, se incluye una página que agrupa los paquetes de acuerdo a las tareas que realizan. Los ítemes que están disponibles incluyen por ejemplo: Inferencia Bayesiana, Análisis de Agrupamiento, Gráficas y Análisis de Series Temporales. La dirección de la página es `http://cran.r-project.org/web/views/`.

Introduciendo órdenes o códigos

Hay muchas formas de obtener ayuda con `R` y `RStudio`. Este sitio, `https://rseek.org/`, me ha servido muchísimo para conocer que paquetes existen para trabajar en un tema específico. Otra forma es preguntar directamente en la consola con las siguientes órdenes:

```
help.start()
help.search("mean")
RSiteSearch("mean")
?mean
help(mean)
```

```
apropos("mean", mode ="function")
```

```
##  [1] ".colMeans"     ".rowMeans"     "colMeans"      "kmeans"
##  [5] "mean"          "mean.Date"     "mean.default"  "mean.difftime"
##  [9] "mean.POSIXct"  "mean.POSIXlt"  "rowMeans"      "weighted.mean"
```

```
example(mean)
```

```
##
## mean> x <- c(0:10, 50)
##
## mean> xm <- mean(x)
##
## mean> c(xm, mean(x, trim = 0.10))
## [1] 8.75 5.50
```

Por llevar un orden al trabajar, lo primero que debemos saber es conocer la dirección de la carpeta de trabajo con `getwd`. Si queremos ajustar la carpeta de trabajo a otra dirección usamos `setwd`.

```
# Conocer la dirección de la carpeta de trabajo.
getwd()
```

```
## [1] "C:/cursoR"
```

```
# Ajustar la dirección de la carpeta de trabajo.
setwd("C:/cursoR")
```

Una buena forma de familiarizarse con la consola es hacer cálculos simples. Trabajando de la forma más básica, podremos ingresar sentencias en la consola, como si usáramos R como una calculadora. El símbolo `<-` es el operador de asignación, también puede utilizarse el signo `=`. En `RStudio` puedes presionar `Alt+guión(-)` para escribir `<-`. Se recomienda usar el `R script` para escribir las órdenes para luego enviarlas a la consola donde se ejecutan presionando `Run` en el menú o las teclas `crtl+Enter` en `RStudio` o las teclas `crtl+R` en `RGUI`. Puedes usar el símbolo de *hashtack* `#` (almohadilla o numeral) para tomar apuntes dentro del `script`, esas líneas con `#` no se ejecutan. Oprime `crtl+L` si quieres limpiar la consola.

```
# Asigno el valor de 35 a la variable x.
# se crea un objeto llamado x tipo numérico
# en el Entorno Global
x <- 35
print(x)
```

```
## [1] 35
```

Analicemos las siguientes líneas de código. Escribe los siguientes comandos en el panel de la consola.

```
msg <- "Hola"
msg
```

```
## [1] "Hola"
```

```
# Adición.
23 + 38
```

```
## [1] 61
```

```
# Sustracción.
57-13
```

```
## [1] 44
```

```
# Multiplicación.
23*72
```

```
## [1] 1656
```

```
2*(3+5)
```

```
## [1] 16
```

```
3*pi
```

```
## [1] 9.424778
```

```
# División.
71/3
```

```
## [1] 23.66667
```

```
534%/%7  # Devuelve el cociente.
```

```
## [1] 76
```

```
534%%7   # Devuelve el residuo.
```

```
## [1] 2
```

```
# Raíz cuadrada.
sqrt(121)
```

```
## [1] 11
```

```
sqrt(2)
```

```
## [1] 1.414214

# Produce máximo 22 dígitos.
print(sqrt(2), 22); print(sqrt(2), 2)

## [1] 1.4142135623730951

## [1] 1.4

# Raíz cuadrada de 2, incluyendo texto en la salida.
cat("La raíz cuadrada de 2 es", sqrt(2),"\n")

## La raíz cuadrada de 2 es 1.414214

# Potencias.
3^5

## [1] 243

9^(1/2)

## [1] 3

# Logaritmo natural.
log(3)

## [1] 1.098612

# Logaritmo de 3 (base 10), o use log10(3).
log(3,10)

## [1] 0.4771213

# Esto es e^3
exp(3)

## [1] 20.08554

# Factorial.
factorial(4)

## [1] 24

# Coseno de 60 grados
cos(60*pi/180)

## [1] 0.5

# Sobre la cantidad de decimales que devuelve
round(2.35719, 2); round(sqrt(2), 3)
```

```
## [1] 2.36
```

```
## [1] 1.414
```

```
round(digits=3, sqrt(2))
```

```
## [1] 1.414
```

```
floor(6.3539); ceiling(6.3539); trunc(6.3539); round(6.3539)
```

```
## [1] 6
```

```
## [1] 7
```

```
## [1] 6
```

```
## [1] 6
```

```
floor(-6.3539); ceiling(-6.3539); trunc(-6.3539); round(-6.3539)
```

```
## [1] -7
```

```
## [1] -6
```

```
## [1] -6
```

```
## [1] -6
```

```
# Números imaginarios
complex(real=1,imaginary=2/3)
```

```
## [1] 1+0.666667i
```

```
(2+5i)*(3+7i)
```

```
## [1] -29+29i
```

```
sqrt(as.complex(-3))
```

```
## [1] 0+1.732051i
```

```
sqrt(2+3i)
```

```
## [1] 1.674149+0.895977i
```

```
# (3+i)*(2-i) Debe llevar el coeficiente 1.
(3+1i)*(2-1i)
```

```
## [1] 7-1i
```

```
# Creando vectores con (:).
1:10
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
#
2.3:11.5
```

```
##  [1]  2.3  3.3  4.3  5.3  6.3  7.3  8.3  9.3 10.3 11.3
```

```
5:-4
```

```
##  [1]  5  4  3  2  1  0 -1 -2 -3 -4
```

```
-(3:12)
```

```
##  [1]  -3  -4  -5  -6  -7  -8  -9 -10 -11 -12
```

```
# Creando vectores usando la función rep().
rep(1, 6)
```

```
## [1] 1 1 1 1 1 1
```

```
rep("Palma", 5) # Las palabras, siempre entre comillas.
```

```
## [1] "Palma" "Palma" "Palma" "Palma" "Palma"
```

```
rep(c(1,2,3), times=5)
```

```
##  [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
```

```
rep(c(1,2,3), each=5)
```

```
##  [1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
```

```
rep(c(1,2,3,4), c(2,3,4,5))
```

```
##  [1] 1 1 2 2 2 3 3 3 3 4 4 4 4 4
```

```
# Creando vectores con la función seq().
seq(10)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
seq(from = 3, to = 35, by =4.5)
```

```
## [1]  3.0  7.5 12.0 16.5 21.0 25.5 30.0 34.5
```

```
seq(50, 24, -3.5)
```

```
## [1] 50.0 46.5 43.0 39.5 36.0 32.5 29.0 25.5
```

```
seq(2, 10, length.out=6)
```

```
## [1]  2.0  3.6  5.2  6.8  8.4 10.0
```

```
seq(2, by=0.5, length.out=8)
```

```
## [1] 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5
```

Oprime `crtl+L` si quieres limpiar la consola.

```
# Combinando 5 números en un vector.
c(3.0, 4.3, 2.5, 3.4, 3.6)
```

```
## [1] 3.0 4.3 2.5 3.4 3.6
```

```
# Almacenando diez números en la variable "Datos".
Datos = c(1.8, 2.5, 3.0, 4.3, 3.5, 2.5, 3.4, 2.5, 3.1, 3.6)
print(Datos)
```

```
##  [1] 1.8 2.5 3.0 4.3 3.5 2.5 3.4 2.5 3.1 3.6
```

```
# Dice cuántos números hay en la variable "Datos".
length(Datos)
```

```
## [1] 10
```

```
# Suma todos los valores de la variable "Datos".
sum(Datos)
```

```
## [1] 30.2
```

```
# Divide cada valor en "Datos" por 2.
Datos/2
```

```
##  [1] 0.90 1.25 1.50 2.15 1.75 1.25 1.70 1.25 1.55 1.80
```

```
# Estadísticas descriptivas para la variable "Datos".
summary(Datos)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.800   2.500   3.050   3.020   3.475   4.300
```

Estadística descriptiva

La estadística descriptiva se encarga de organizar, describir y presentar los datos en forma concisa. La forma más común de describir un conjunto de datos relacionados entre sí es reportar un valor medio y una dispersión alrededor de dicho valor medio. En los datos de muestreos y de población total (censo), una medida de tendencia central proporciona una evaluación de un promedio de los datos. Una medida de dispersión, o de variabilidad, es una indicación de la extensión de las medidas alrededor del centro de la distribución. También, hay medidas de posición como los cuartiles y percentiles.

La **media** es la medida básica para describir el valor central de un conjunto de datos.

La media se define como:

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

La segunda medida de valor central de un conjunto de datos es la **mediana**, definida como el valor en el medio cuando los datos son ordenados de menor a mayor. La mediana es uno de los valores que componen el conjunto, no proviene de ningún cálculo, sino de observar el valor dentro del conjunto después de haberlo ordenado. La moda es otra forma de describir el valor central de un conjunto de datos y representa el valor más frecuente. El lenguaje R no proporciona una función para obtener la moda.

Para poder describir mejor un conjunto de datos necesitamos una medida de dispersión ademas de una del valor central, la más simple es el rango, el cual muestra los valores mínimo y máximo del conjunto de datos. La varianza y la desviación estándar son las medidas de dispersión más conocidas. La **varianza** es la esperanza del cuadrado de la desviación típica de dicha variable respecto a su media. La **desviación estándar** es la raíz cuadrada de la varianza. Una propiedad útil de la **desviación estándar** es que, a diferencia de la varianza, se expresa en las mismas unidades que los datos a partir de los que se calcula.

La varianza se define como:

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

Los **cuartiles** son los tres valores de la variable que dividen a un conjunto de datos ordenados en cuatro partes iguales. Los **percentilos** son los 99 valores que dividen la serie de datos en 100 partes iguales.

La desviación estándar se define como:

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

Análisis de estadística descriptiva

Datos de temperatura corporal humana

Para poner en contexto los conceptos de la estadística descriptiva, analizaremos el `data set` *HumanBodyTemp* del paquete `abd` (Middleton & Pruim, 2015). Recuerda instalarlo antes. *HumanBodyTemp* es un `dataframe` con 25 observaciones y una variable, `temp`, con valores de temperatura corporal en grados *Fahrenheit*. La temperatura corporal se midió a 25 personas sanas elegidas al azar.

```
install.packages("abd", dep=TRUE)
```

Llamamos el paquete `abd` con la función `library` para incorporarlo a la sesión de `R` y tener a disposición sus `data sets` y ejemplos de códigos (`abd` requiere paquetes como: `lattice` , `mosaic` , `dplyr` (Wickham et al., 2021) y otros). Ahora, creamos una copia del objeto con el nombre `TempCuerpoHumano`.

El paquete complementario de `lattice` es una implementación de paneles gráficos para R. El paquete `lattice` es un sistema de visualización de datos de alto nivel, potente y elegante, con énfasis en datos multivariados (Sarkar, 2008).

El paquete de `mosaic` proporciona una introducción simplificada y sistemática a la funcionalidad principal relacionada con la estadística descriptiva, la visualización, el modelado y la inferencia basada en simulación (Pruim et al., 2017).

```
library(abd)
data("HumanBodyTemp")
TempCuerpoHumano <- HumanBodyTemp
```

Observar los datos

La función `head` muestra los datos iniciales de la serie dada sin ordenar.

```
# temperatura corporal en grados Fahrenheit.
head(TempCuerpoHumano)
```

```
##   temp
## 1 98.4
## 2 98.6
## 3 97.8
## 4 98.8
## 5 97.9
## 6 99.0
```

También, con la función `tail` se muestran los datos finales de la serie dada sin ordenar.

```
# temperatura corporal en grados Fahrenheit.
tail(TempCuerpoHumano)
```

```
##     temp
## 20  97.5
## 21  97.5
## 22  98.8
## 23  98.6
## 24 100.0
## 25  98.4
```

Y con la función `str` se muestra que tipo de objeto es nuestra base de datos, cual es su dimensión o sea cuantas observaciones y variables posee, y presenta los tipos de variable y las primeras observaciones.

```
str(TempCuerpoHumano)
```

```
## 'data.frame':    25 obs. of  1 variable:
##  $ temp: num  98.4 98.6 97.8 98.8 97.9 99 98.2 98.8 98.8 99 ...
```

Ordenar los datos

Con la función `sort` se ordenan los valores en forma ascendente o descendente.

```
sort(TempCuerpoHumano$temp)
```

```
##  [1]  97.4  97.5  97.5  97.6  97.8  97.9  98.0  98.2  98.4  98.4  98.4  98.4
## [13]  98.6  98.6  98.8  98.8  98.8  98.8  99.0  99.0  99.1  99.2  99.4  99.5
## [25] 100.0
```

```
sort(TempCuerpoHumano$temp, decreasing = TRUE)
```

```
##  [1] 100.0  99.5  99.4  99.2  99.1  99.0  99.0  98.8  98.8  98.8  98.8  98.6
## [13]  98.6  98.4  98.4  98.4  98.4  98.2  98.0  97.9  97.8  97.6  97.5  97.5
## [25]  97.4
```

Funciones básicas

Contamos con varias funciones básicas para obtener estadísticas descriptivas de nuestros datos: `min` = mínimo (devuelve el valor mínimo de todos los valores presentes en sus argumentos), `max` = máximo (devuelve el valor máximo de todos los valores presentes en sus argumentos), `range` = rango (devuelve un vector que contiene el mínimo y el máximo de todos los argumentos dados), `mean` = media (devuelve la media aritmética, o sea, la suma de una serie de números dividida por la cuenta de números en la serie), `median` = mediana (devuelve el valor que se encuentra en el medio de la serie de datos cuando esta se ordenada de menor a mayor), `var` = varianza (devuelve la esperanza del cuadrado de la desviación típica de dicha variable respecto a su media), `sd` = desviación estándar (devuelve la raíz cuadrada de la varianza), `fivenum` = cinco números (devuelve el valor mínimo, los tres valores de la variable que dividen a el conjunto de datos ordenados en cuatro partes iguales, y el valor máximo), `quantile` = percentilos (de manera predeterminada devuelve los percentilos 0 %, 25 %, 50 %, 75 % y 100 %, que corresponden a el valor mínimo, los cuartiles y el valor máximo; también se puede conocer cualquiera de los 99 percentilos) y `summary` = resumen (devuelve un resumen de las medidas descriptivas).

```
mean(TempCuerpoHumano$temp)   # Media
```

```
## [1] 98.524
```

```
median(TempCuerpoHumano$temp)     # Mediana
```

```
## [1] 98.6
```

```
table(TempCuerpoHumano$temp)      # Tabla de frecuencias
```

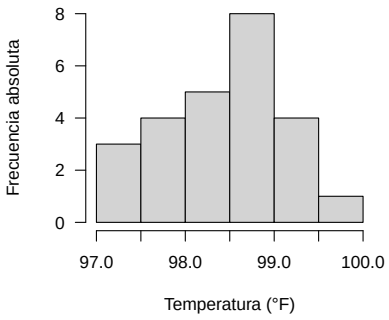


Figura 1. Histograma con la función hist.

```
##
## 97.4 97.5 97.6 97.8 97.9   98 98.2 98.4 98.6 98.8   99 99.1 99.2 99.4 99.5  100
##    1    2    1    1    1    1    1    4    2    4    2    1    1    1    1    1
```

```
                              # de los valores
which.max(table(TempCuerpoHumano$temp)) # Moda
```

```
## 98.4
##    8
```

```
sd(TempCuerpoHumano$temp)        # Desviación estándar
```

```
## [1] 0.6777905
```

```
var(TempCuerpoHumano$temp)      # Varianza
```

```
## [1] 0.4594
```

```
range(TempCuerpoHumano$temp)    # Rango
```

```
## [1]  97.4 100.0
```

```
min(TempCuerpoHumano$temp)      # Mínimo
```

```
## [1] 97.4
```

```
max(TempCuerpoHumano$temp)      # Máximo
```

```
## [1] 100
```

```
fivenum(TempCuerpoHumano$temp) # cuartiles
```

```
## [1]  97.4  98.0  98.6  99.0 100.0
```

```
quantile(TempCuerpoHumano$temp)# cuartiles
```

```
##    0%   25%   50%   75%  100%
##  97.4  98.0  98.6  99.0 100.0
```

```
quantile(TempCuerpoHumano$temp,0.15)    # percentilo 15
```

```
##   15%
## 97.72
```

```
quantile(TempCuerpoHumano$temp,0.95)    # percentilo 95
```

```
##   95%
## 99.48
```

```
summary(TempCuerpoHumano$temp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   97.40   98.00   98.60   98.52   99.00  100.00
```

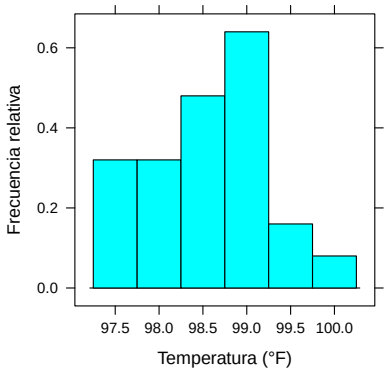


Figura 2. Histograma con la función histogram.

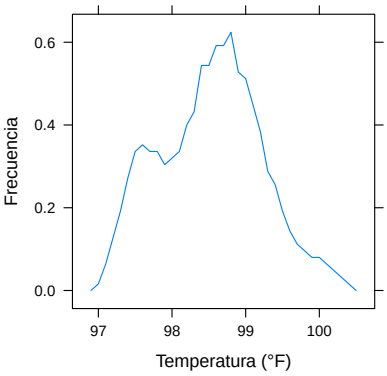


Figura 3. Polígono de frecuencia con la función ashplot.

Representación gráfica

Histograma

Las tablas de frecuencia se pueden visualizar en un gráfico llamado Histograma con la función `hist` del paquete `graphics` o con `histogram` del paquete `lattice`. La representación gráfica más popular de un conjunto de datos es el histograma, el cual representa la frecuencia de aparición de valores dentro del rango del conjunto de datos.

```
hist(TempCuerpoHumano$temp, las=1, xlab = "Temperatura (°F)",
   breaks = 5, ylab = "Frecuencia absoluta", main = " ")

histogram(~temp, TempCuerpoHumano, xlab = "Temperatura (°F)",
   width=0.5, ylab = "Frecuencia relativa", type = "density")
```

Otros gráficos

Otros gráficos para visualizar la distribución de las frecuencias de los datos. Las funciones que usaremos pertenecen al paquete `mosaic`: `ashplot`, `densityplot` y `dotPlot`.

```
ashplot(~temp, TempCuerpoHumano, xlab = "Temperatura (°F)",
   width = 0.5, ylab = "Frecuencia")


densityplot(~temp, TempCuerpoHumano,
 xlab = "Temperatura (°F)",    width=0.5, ylab = "Frecuencia")

dotPlot(~temp, data=TempCuerpoHumano,
 xlab = "Temperatura (°F)", width=1, ylab = "Frecuencia")
```

Diagrama de cajas

El diagrama de cajas es la forma gráfica de comunicar los cinco valores que describen de forma concisa un conjunto de datos, estos valores son: el mínimo, el percentilo 25, la mediana (percentilo 50), el percentilo 75, el máximo.

```
boxplot(TempCuerpoHumano$temp,
 ylab = "Temperatura (°F)", main = " ", las = 1)


bwplot(~temp, data = TempCuerpoHumano,
 xlab = "Temperatura (°F)", pch = "|")
```

Gráfico de rama y hoja

Este gráfico lo podemos obtener con la función `stem` del paquete `graphics`. Los números que se muestran a la izquierda del carácter `|` son los dígitos mas significativos. Es una forma ordenada y simplificada de ver todos los valores del conjunto de datos.

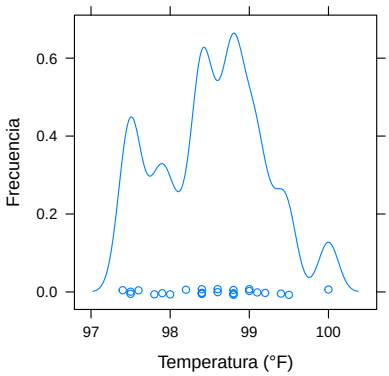


Figura 4. Polígono de frecuencia con la función densityplot.

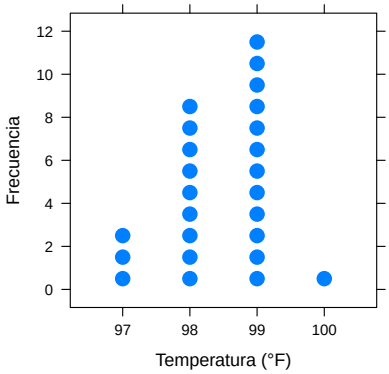


Figura 5. Polígono de frecuencia con la función dotPlot.

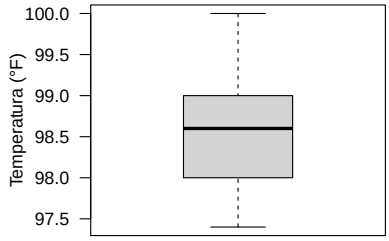


Figura 6. Diagrama de cajas con la función boxplot.

```
stem(TempCuerpoHumano$temp, scale = 2)
```

```
##
##   The decimal point is at the |
##
##    97 | 4
##    97 | 55689
##    98 | 024444
##    98 | 668888
##    99 | 00124
##    99 | 5
##   100 | 0
```

Otras funciones

La función `favstats` del paquete `mosaic` nos muestra un resumen de las medidas descriptivas.

```
favstats(TempCuerpoHumano$temp)
```

```
##   min Q1 median Q3 max   mean        sd  n missing
##  97.4 98   98.6 99 100 98.524 0.6777905 25       0
```

La función `stat.desc()` del paquete `pastecs` nos devuelve un resumen de las medidas descriptivas de un conjunto de datos (Grosjean & Ibáñez, 2018).

```
library(pastecs)
```

```
options(scipen=999)
round(stat.desc(TempCuerpoHumano), 2) # Paquete pastecs
```

```
##                 temp
## nbr.val        25.00
## nbr.null        0.00
## nbr.na          0.00
## min            97.40
## max           100.00
## range           2.60
## sum          2463.10
## median         98.60
## mean           98.52
## SE.mean         0.14
## CI.mean.0.95    0.28
## var             0.46
## std.dev         0.68
## coef.var        0.01
```

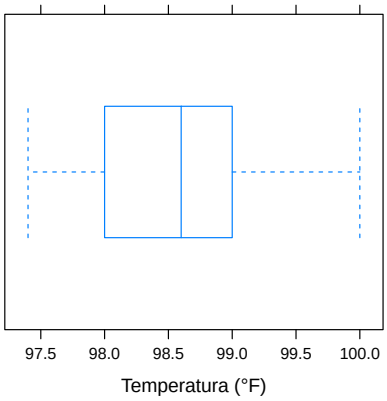


Figura 7. Diagrama de cajas con la función bwplot.

Ajustamos la opción `options(scipen=999)` para que los resultados no se impriman en notación científica.

La función `describe` del paquete `Hmisc` nos devuelve un resumen de las medidas descriptivas de un conjunto de datos (Harrell Jr et al., 2021).

```
library(Hmisc)

describe(TempCuerpoHumano)  # Paquete Hmisc
```

```
## TempCuerpoHumano
##
##  1  Variables      25  Observations
## ---------------------------------------------------------------------------
## temp
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##       25        0       16    0.991    98.52   0.7833    97.50    97.54
##      .25      .50      .75      .90      .95
##    98.00    98.60    99.00    99.32    99.48
##
## lowest :  97.4  97.5  97.6  97.8  97.9, highest:  99.1  99.2  99.4  99.5 100.0
##
## Value       97.4  97.5  97.6  97.8  97.9  98.0  98.2  98.4  98.6  98.8  99.0
## Frequency      1     2     1     1     1     1     1     4     2     4     2
## Proportion  0.04  0.08  0.04  0.04  0.04  0.04  0.04  0.16  0.08  0.16  0.08
##
## Value       99.1  99.2  99.4  99.5 100.0
## Frequency      1     1     1     1     1
## Proportion  0.04  0.04  0.04  0.04  0.04
## ---------------------------------------------------------------------------
```

Acerca de los autores

José Ulises Jiménez S.

https://orcid.org/0000-0003-1302-5269
Biólogo con especialidad en Botánica, graduado de la Universidad de Panamá; cuenta además, con una Maestría Científica en Manejo y Conservación de Bosques Tropicales y Biodiversidad del Centro Agronómico Tropical de Investigación y Enseñanza (CATIE) de Costa Rica. Labora como investigador en el Centro de Investigaciones Hidráulicas e Hidrotécnicas y un apasionado usuario de R y LaTeX; tiene una vasta experiencia en caracterización de la línea base de los factores biológicos, principalmente en inventarios forestales. Como docente tiempo parcial de la Universidad Tecnológica de Panamá ha dictado cursos de Ecología General, Introducción a la Evaluación de Impacto Ambiental y Práctica de Campo.

José Rogelio Fábrega D.

https://orcid.org/0000-0003-1536-0386
Ingeniero Civil graduado de la Universidad Santa María La Antigua, con Maestría y Doctorado en Ingeniería Civil con énfasis en Ingeniería Ambiental de la Universidad de Purdue, Estados Unidos. Ha sido investigador por más de 15 años en diversas áreas de la Ingeniería Ambiental y de los Recursos Hídricos. Recientemente, ha trabajado en temas relativos al ciclo del agua y del carbono, manejo de cuencas y Cambio Climático. Ha publicado más de quince artículos en revistas científicas indexadas. Posee idoneidades como Ingeniero Civil en Panamá, Costa Rica y en Indiana, E.U.A. Hoy día, es Director del Centro de Investigaciones Hidráulicas e Hidrotécnicas (CIHH) de la UTP. Ha sido investigador principal y co-investigador en múltiples proyectos financiados por SENACYT, la Organización Internacional de Energía Atómica (OIEA) y JICA. Ha dictado clases en la USMA y UTP. Desde el 2013 ha sido presidente del capítulo de Panamá del *Global Water Partnership* (GWP), y representante de APANAC ante la Red Interamericana de Academias de Ciencias (IANAS).

Referencias

Allaire, J., Xie, Y., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., & Iannone, R. (2021). *rmarkdown: Dynamic Documents for R*. R package version 2.8.

Grosjean, P. & Ibáñez, F. (2018). *pastecs: Package for Analysis of Space-Time Ecological Series*. R package version 1.3.21.

Harrell Jr, F. E., with contributions from Charles Dupont, & many others. (2021). *Hmisc: Harrell Miscellaneous*. R package version 4.5-0.

Middleton, K. M. & Pruim, R. (2015). *abd: The Analysis of Biological Data*. R package version 0.2-8.

Pruim, R., Kaplan, D. T., & Horton, N. J. (2017). The mosaic package: Helping students to 'think with data' using r. *The R Journal*, 9(1), 77–102.

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

RStudio Team (2021). *RStudio: Integrated Development Environment for R*. RStudio, PBC, Boston, MA.

Sarkar, D. (2008). *Lattice: Multivariate Data Visualization with R*. New York: Springer. ISBN 978-0-387-75968-5.

Wickham, H., François, R., Henry, L., & Müller, K. (2021). *dplyr: A Grammar of Data Manipulation*. R package version 1.0.6.

Xie, Y. (2014). knitr: A comprehensive tool for reproducible research in R. In V. Stodden, F. Leisch, & R. D. Peng (Eds.), *Implementing Reproducible Computational Research*. Chapman and Hall/CRC. ISBN 978-1466561595.

Xie, Y. (2015). *Dynamic Documents with R and knitr*. Boca Raton, Florida: Chapman and Hall/CRC, 2nd edition. ISBN 978-1498716963.

Xie, Y. (2021). *knitr: A General-Purpose Package for Dynamic Report Generation in R*. R package version 1.33.

Xie, Y. & Allaire, J. (2020). *tufte: Tufte's Styles for R Markdown Documents*. R package version 0.9.

Xie, Y., Allaire, J., & Grolemund, G. (2018). *R Markdown: The Definitive Guide*. Boca Raton, Florida: Chapman and Hall/CRC. ISBN 9781138359338.

Xie, Y., Dervieux, C., & Riederer, E. (2020). *R Markdown Cookbook*. Boca Raton, Florida: Chapman and Hall/CRC. ISBN 9780367563837.