

# Introducción al Análisis de Datos Geoespaciales

Horacio Samaniego, Derek Corcoran, Giorgia Graells

2023-08-14



# Contents

<b>1</b>	<b>Datos del curso</b>	<b>5</b>
1.1	DESCRIPCIÓN DEL CURSO . . . . .	5
1.2	Evaluación . . . . .	6
1.3	Calendario . . . . .	7
1.4	Recursos adicionales . . . . .	7
<b>2</b>	<b>Introducción</b>	<b>9</b>
2.1	Objetos . . . . .	9
2.2	Variables . . . . .	10
2.3	Funciones . . . . .	10
2.4	Vectores . . . . .	12
2.5	Instalar librerías . . . . .	13
2.6	R Notebook . . . . .	13
2.7	Leer datos . . . . .	14
2.8	Ejercicios . . . . .	14



# Chapter 1

## Datos del curso

- **Universidad Austral de Chile:** Laboratorio de Ecoinformática
- **Nombre asignatura:** Introducción al Análisis de Datos Geoespaciales
- **Código asignatura:** CBIT200
- **Docente responsable:** Horacio Samaniego
- **Correo electrónico:** horaciosamaniego@uach.cl
- **Creadores:** Derek Corcoran B. & Giorgia Graells C.
- **Modalidad de clases:**
  - Prácticas
  - Presenciales
  - Consultas por Discord (chat o video) – link válido hasta 9/9/2023
- **Horario de clases:** Jueves 10:00 - 13:00 hrs
- **Lugar:** Sala de computación, Facultad de Ciencias Forestales y Recursos Naturales, Campus Isla Teja, Valdivia
- **Inicio clases:** 10 agosto 2023
- **Término clases:** 30 noviembre 2023 (puede modificarse según calendario académico)
- **Web del curso** [aquí](#)

### 1.1 DESCRIPCIÓN DEL CURSO

Este curso tiene como objetivo central adquirir herramientas para el manejo de datos con un énfasis en datos espaciales para el manejo de los recursos naturales y la aproximación y resolución de problemas ambientales. Se busca la creación de competencias en los principios de investigación reproducible, representación y análisis de información espacial y la creación de mapas estáticos e interactivos. Esto permitirá la adquisición de herramientas para profundizar el conocimiento acerca del diseño y desarrollo de análisis de datos ambientales complejos y espacialmente explícitos.

### 1.1.1 OBJETIVOS

1. Conocer y entender el concepto de Investigación Reproducible como una forma y filosofía de trabajo que permite que las investigaciones sean más ordenadas y replicables, desde la toma de datos hasta la escritura de resultados utilizando R
2. Realizar análisis críticos de la naturaleza de los datos al realizar análisis exploratorios y reforzar conocimientos en estadística
3. Realizar análisis de datos espaciales, poder hacer mapas y aplicar a preguntas de conservación y manejo de recursos naturales.
4. Aprender a utilizar de forma proficientemente el lenguaje de programación R y la plataforma GitHub en un ambiente de trabajo colaborativo.

## 1.2 Evaluación

### 1.2.1 tareas

- Se entregarán ejercicios que deben resolverse semanales. La entrega se hará usando la plataforma GitHub. Cada estudiante será responsable de entregar su tarea y de corregir a tres compañeros al azar. Las correcciones ocurrirán en modalidad “doble ciego”, es decir, tanto el corrector como el autor de la tarea serán mantenidos como anónimo. Esta modalidad de corrección por pares será moderada por el profesor y seguirá una pauta entregada para cada tarea. Los estudiantes recibirán un punto por cada tarea revisada, lo que se sumará para completar el puntaje de su propia tarea.

### 1.2.2 proyecto

- Se desarrollará un proyecto de análisis y de programación que será desarrollado durante el último mes de clases y presentado en las últimas dos sesiones del curso.

### 1.2.3 calificaciones

<i>Evaluación</i>	<i>Ponderación</i>
Ejercicios & Tareas $\frac{1}{n} \sum_i nota\ tarea_i$	50%
Proyecto reporte	10%
Proyecto código	20%
Presentación	15%
Participación / Asistencia	5%

## 1.3 Calendario

Semana	Fecha	Actividad
1	10 Agosto	<ul style="list-style-type: none"> <li>• Presentación del curso</li> <li>• Procedimientos y reglas de evaluación.</li> </ul>
2	17 Agosto	<ul style="list-style-type: none"> <li>• R, sus variables, objetos y funciones</li> <li>• Introducción a GitHub;</li> <li>• Investigación reproducible</li> </ul>
3	24 Agosto	Markdown y Rmd avanzado
4	31 Agosto	Visualización I <ul style="list-style-type: none"> <li>• ggplot2</li> <li>• otras librerías</li> </ul>
5	7 Septiembre	Correlación y Regresión
6	14 Septiembre	Introducción a R como un Sistema de
7	21 Septiembre	Información Geográfica
8	28 Septiembre	Rasters
9	5 Octubre	Modelos en rasters
10	12 Octubre	Autocorrelación espacial
11	19 Octubre	Visualización II (mapas)
12	26 Octubre	Visualización III (interactiva)
13	2 Noviembre	Sensores Remotos
14	9 Noviembre	Proyecto - Definición de Objetivos
15	16 Noviembre	Proyecto - Modelo de Estudio e hipótesis de trabajo
16	23 Noviembre	Proyecto - Resultados
17	30 Noviembre	Proyecto - Presentación de Proyectos
		Epílogo & Presentación de Proyectos

## 1.4 Recursos adicionales

Si bien intentamos buscar ejemplos originales y sets de datos locales, gran parte del material con que trabajaremos ha sido ya trabajado elaborado por otros. Es por eso que se sugiere revisar algunos sitios claves como los siguientes:

- R for Data Sciences
- Stackoverflow
- [GIS Stack Exchange] (<https://gis.stackexchange.com/>)
- [Spatial Data Science] (<https://rspatial.org/index.html>)





## Chapter 2

# Introducción

R es un entorno y lenguaje de programación con un enfoque al análisis estadístico. Permite hacer todos los análisis numéricos que requieras en tu vida profesional. Es una implementación de libre distribución de otro programa estadístico de uso comercial, S. Al ser software libre, es la comunidad de usuarios la que guía su desarrollo, transformándolo en uno de los programas más versátiles para trabajos cuantitativos existentes hoy en día. La página principal desde la que se puede acceder a los archivos y documentación necesarias para su utilización es:

[www.r-project.org](http://www.r-project.org)

Si bien R es un software que puede usarse desde la línea de comando, para trabajar utilizaremos <http://www.rstudio.org>.

Este es un Entorno de Desarrollo Integrado (IDE, por su sigla en inglés) que, al igual que R, es software libre y permite integrar herramientas necesarias para el desarrollo y así facilitarlo. La página oficial para descargarlo es:

[www.rstudio.com](http://www.rstudio.com)

## 2.1 Objetos

En términos genéricos, todos los elementos que R maneja son objetos. Un objeto tiene ciertas propiedades y en ocasiones es capaz de llevar a cabo ciertas tareas si se le dan los argumentos necesarios. Por ejemplo, un teléfono es capaz de realizar llamadas siempre que le demos el número a marcar.

## 2.2 Variables

Al momento de trabajar, es probable que necesitemos guardar valores o cálculos, de manera que no necesitemos escribirlos cada vez que los usemos, para esto utilizamos variables.

Para realizar una asignación de variable:

```
a = 200
```

Luego, podemos utilizar el valor contenido en la variable, utilizando su nombre:

```
print(a)
```

```
## [1] 200
```

### 2.2.1 Tipos de variables

Existen diversos tipos o clases de variables, dependiendo de las características del objeto que les es asignado. Para conocer a qué tipo corresponde un objeto usamos `class`:

```
x=7  
x
```

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

```
class (x)
```

```
## [1] "numeric"
```

```
x=5/3  
x
```

```
## [1] 1.666667
```

```
class (x)
```

```
## [1] "numeric"
```

```
x="Trece"  
x
```

```
## [1] "Trece"
```

```
class (x)
```

```
## [1] "character"
```

## 2.3 Funciones

Muchas cosas en R pueden hacerse a través del uso de funciones, estas permiten realizar operaciones típicas sin necesidad de escribir grandes cantidades

de código. Por ejemplo:

```
sqrt(10)
```

```
## [1] 3.162278
```

```
round(1.9)
```

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

```
seq(0,10)
```

```
## [1] 0 1 2 3 4 5 6 7 8 9 10
```

```
seq(0,10,2)
```

```
## [1] 0 2 4 6 8 10
```

```
rep(5,10)
```

```
## [1] 5 5 5 5 5 5 5 5 5 5
```

```
paste(seq(5,10), "elefantes")
```

```
## [1] "5 elefantes" "6 elefantes" "7 elefantes" "8 elefantes" "9 elefantes"
```

```
## [6] "10 elefantes"
```

Los datos o variables que van dentro de las funciones, se denominan *argumentos* y cada función requiere que se le entreguen los argumentos apropiados para ejecutar la acción prevista.

Por ejemplo, la función `mean()` no puede calcular el promedio si como argumentos se le pasan letras.

```
mean(c("a", "b", "c"))
```

```
## Warning in mean.default(c("a", "b", "c")): argument is not numeric or logical:
```

```
## returning NA
```

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

Esto es importante, porque al introducir datos podemos estar utilizando números como palabras:

```
1, 2, 3 "1", "2", "3"
```

Si nos encontramos con este problema, debemos transformar los datos al tipo o clase adecuada, con las funciones:

```
as.numeric() y as.character()x'
```

## 2.4 Vectores

Conjunto ordenado de valores del mismo tipo, agrupados en un único objeto. Para crear una variable vector utilizamos:

```
v = c(1,1,2,3)
vector = c("mi", "primer", "vector")
vector
```

```
## [1] "mi"      "primer" "vector"
```

Cada objeto dentro de un vector posee un índice, el cual indica la posición que ocupa dentro del vector, para acceder a una posición específica usamos:

```
vector[1]
```

```
## [1] "mi"
```

```
vector[2]
```

```
## [1] "primer"
```

```
vector[3]
```

```
## [1] "vector"
```

y si queremos reemplazar alguno de esos objetos:

```
vector[2]="segundo"
vector
```

```
## [1] "mi"      "segundo" "vector"
```

Un vector permite almacenar varios valores en una única variable y permite ejecutar operaciones o funciones a un conjunto de datos:

```
vector = c(1,2,3,4,5)
vector*2
```

```
## [1]  2  4  6  8 10
```

```
vector^2
```

```
## [1]  1  4  9 16 25
```

o incluso realizar operaciones entre vectores:

```
v1=c(1:3)
v2=c(6,8,10)
```

```
v1
```

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

```
v2
```

```
## [1] 6 8 10
```

```
v1 + v2
```

```
## [1] 7 10 13
```

```
v1*v2
```

```
## [1] 6 16 30
```

```
v3=c("a","b","c")
```

```
v1 * v3
```

```
## Error in v1 * v3: non-numeric argument to binary operator
```

## 2.5 Instalar librerías

Muchas veces las funciones incorporadas en R son insuficientes para nuestros fines, por lo que es necesario instalar paquetes o “packages” de herramientas hechas por la comunidad. En este caso, usaremos el paquete “openxlsx”, que nos permite leer archivos Excel. Para instalarlo:

```
install.packages("openxlsx")
```

Debe hacerse una única vez, los paquetes quedan instalados en nuestra versión de R. Y para usarlo dentro de nuestro proyecto:

```
library(openxlsx)
```

Debe incluirse en cada proyecto donde queramos usarlo y ejecutarse cada vez que abrimos R.

## 2.6 R Notebook

Un Notebook en R es un documento con bloques o “chunks” que pueden ser ejecutados directa e interactivamente, para así visualizar los resultados directamente bajo el código.

Para instalar esta librería:

```
install.packages("rmarkdown")
```

Una vez instalada, puedes crear un nuevo notebook en RStudio llenando a *File -> new file -> R notebook*.

Agrega un nuevo chunk haciendo click en el botón *Insert Chunk* en la barra de herramientas o presionando *Ctrl+Alt+I*

Un chunk puede ser ejecutado usando:

1. Haciendo click en el triángulo verde o “Run Current Chunk” en la esquina superior derecha de cada chunk.

2. Clickeando al interior de un chunk y presionando *Ctrl + Enter*.

De ambas formas se ejecutará todo el código contenido dentro de el chunk.

Cuando guardas el notebook, un archivo HTML que contiene el código y los resultados se guardará junto a él (Click en el botón de *Preview* o presiona *Ctrl+Shift+K* para previsualizar el archivo HTML)

## 2.7 Leer datos

Delimitados por coma: `read_csv("file.csv")`

Con cualquier delimitador: `read_delim("file.txt", delim = "|")`

## 2.8 Ejercicios

1. Cree una nueva variable que contenga un vector con 10 números aleatorios
2. multiplíquela por seis.
3. cree una segunda variable que contenga una secuencia de 5 caracteres
4. combine las dos variable en una sola variable
5. ¿cuál es el largo de esta última variable creada?
6. ¿de qué tipo es esta variable?
7. ¿qué sucede si divie esta última variable por 3?
8. cree un vector con los elementos 1 2 3 4 5 6 y llámelo `x`
9. cree un nuevo vector con los elementos 10 20 30 40 50 y llámelo `y`
10. ¿qué ocurre si intenta sumar `x` e `y`? explique
11. agregue el valor 60 al vector `y` (hint: you can use the `c()` function)
12. sume `x` e `y`
13. multiplique `x` e `y`
14. cree un `data.frame` con el mínimo código posible usando los datos de la siguiente imagen y llámelo `z`:

Numero	Letra	Medida
5	S	49.59
4	L	91.84
3	B	49.32
1	Q	72.25
2	A	65.51

15. cree un dataframe de datos ficticios que represente una muestra de 100 individuos de aves y su tamaño corporal. Use 4 tipos de aves: un paseriforme, un columbiforme, un rapaz y trochiliformes en aproximadamente las mismas proporciones. (Averigue el real tamaño promedio de cada grupo)
16. grafique los pesos corporales de cada grupo (ayuda: puede usar graficos de distribuciones, caja y bigote o bien de violín)