

# Contents

<b>Ciencia de datos para politologues</b>	<b>3</b>
<b>1 Nociones básicas de R</b>	<b>5</b>
<b>2 (PART*) Análisis espacial</b>	<b>7</b>
<b>3 Introducción al análisis epsacial</b>	<b>9</b>
<b>4 Nociones básicas de vizualización</b>	<b>11</b>
4.1 Vizualizar lo más elemental: una base de datos con coordenadas	11
4.2 EXTRA: Agregando un mapa base con ‘ggmap()’ . . . . .	13



# Ciencia de datos para politologues

---

“*Ciencia de datos para politologues*” es un libro práctico sobre las principales técnicas de análisis de datos en **R**. El libro contiene los conceptos y técnicas básicas tanto estadística como computacionales, que son de indispensables conocimiento para estudiante ciencias políticas, relaciones internacionales y/o gestión y desarrollo urbanos. Por el lado de la estadística, los contenidos de este libro van desde conceptos básicos de estadística descriptiva, hasta el repaso general y práctico de técnicas econométricas de evaluación de impacto y *machine learning*. Por el lado de programación, este libro busca explicar desde el manejo básico de **R**, hasta algunas técnicas de computación en la nube y la creación de algoritmos de extracción de información y análisis de textos (*text-mining*).

## Un poco sobre mi

Mi nombre es Guibor, pero por lo general me dicen “Guibi”. Estudie gestión y desarrollo urbanos en la Facultad de estudios Políticas, Internacionales y Urbanos, de la *Universidad del Rosario* (Colombia), y en esta misma universidad realice mis estudios de maestría en economía y economía de las políticas públicas. En el día a día trabajo como científico de datos y me gusta mucho también trabajar en algunas investigaciones independientes en diversos temas. Aquí un poco más de mi perfil.

## ¿Un libro más de ciencia de datos y R? ¿Para qué?

A lo largo de mi carrera y en mi ejercicio como investigador, me ha costado mucho encontrar una guía que explique ciertas nociones básicas de estadística y programación para personas que no tengan una formación profesional en matemáticas, estadística y/o economía. Por otro lado, también he luchado (y visto muchos luchar) a la hora de trabajar con softwares licenciados y pagos como Office, ArcGIS, Stata y muchos otros, en los que uno se puede ver atrapado en una de dos situaciones: O consigues una licencia paga (muy cara) de manera directa o a través de la universidad (a través de trámites tediosos que

toca anualmente re-hacer), o consigues una licencia pirata exponiendo tu información y tus equipos a cualquier forma de *malware*. Finalmente, si bien hay muy buenos y numerosos recursos en línea, pero están dispersos, muchas veces solo en inglés, o simplemente son requieren de un buen conocimiento previo de matemáticas, estadística y/o programación e ingeniería de datos para entenderlos, o para entender cómo se pueden aplicar al mundo del análisis de políticas públicas y situaciones socio-políticas.

De este modo, este libro busca que cualquier politólogo tenga un muy buen arsenal de herramientas de análisis, y pueda tener (sin importar su conocimiento previo en matemáticas, estadística, programación e ingeniería) una noción básica pero muy robusta de cómo funciona el mundo de la tecnología aplicada a los problemas de investigación del día a día. Y finalmente, el libro busca brindar estos conocimientos a través de aprender el uso de **R**: un lenguaje/software de programación open source, al alcance de cualquiera con un computador de manejar gratuita y libre, y con una de las comunidades más grande de usuarios en el mundo.

“<https://www.youtube.com/watch?v=9i0ElncHGRg>”

guibor.camargo@urosario.edu.co

## Chapter 1

# Nociones básicas de R

Para empezar a utilizar R, es necesario primero descargar el programa, y preferiblemente también descargar la interfaz *Rstudio*. R es un lenguaje de programación, y *Rstudio* es un entorno en el cuál podemos tener más y visualización en nuestra interacción con este lenguaje. Un ejemplo útil de esto es la diferencia que hay entre JavaScript y Google Chrome: A través de JavaScript y http nos comunicamos con internet, pero a través de Google Chrome interactuamos con una experiencia de usuario con estos protocolos y códigos.

### Un lenguaje de programación orientado a objetos

**R** es un *lenguaje de programación orientado a objetos*. Esto quiere decir, que por ejemplo



## Chapter 2

### (PART\*) Análisis espacial





## Chapter 3

# Introducción al análisis espacial

Esta guía se base en el curso de Análisis Espacial de Datacamp, y en algunos recursos disponibles en en línea como [recurso 1](#) y [recurso 2](#). Esta guía asume que el lector ya esta familiarizado con:

- Manejo básico de R
- Manejo de las librerías `dplyr` y `ggplot2`
- Conceptos básicos de análisis espacial

De esta manera, no se hará mucho énfasis en explicar el trasfondo estadístico de algunas de las técnicas aquí explicadas. Los recursos geográficos y tabulares empleados en esta guía pueden consultarse en el [este drive](#).

### Acrónimos:

- CRS: *Coordinate Reference Systems* (sistemas de coordenadas)

### Librerías previas

```
library(dplyr) # Manejo de bases de datos
library(tidyr) # Manejo de bases de datos
library(ggplot2) # Creación de gráficas
library(googledrive) # Conexión con google drive (para descargar datos de pruebas)
```

Antes de iniciar, cargaremos la base de homicidios de Bogotá en 2005-2019. Podemos descargarla manualmente del [drive](#) de la guía usar el siguiente código.

```
# Cargar base de homicidios de dreive
{drive_download(as_id(drive_ls("https://drive.google.com/drive/folders/10VeNIEctH2rq7T
a01_homicidios <- read.csv("01_tabla_homicidios_bog_2005_2019.csv")
file.remove("01_tabla_homicidios_bog_2005_2019.csv")}
```

## Chapter 4

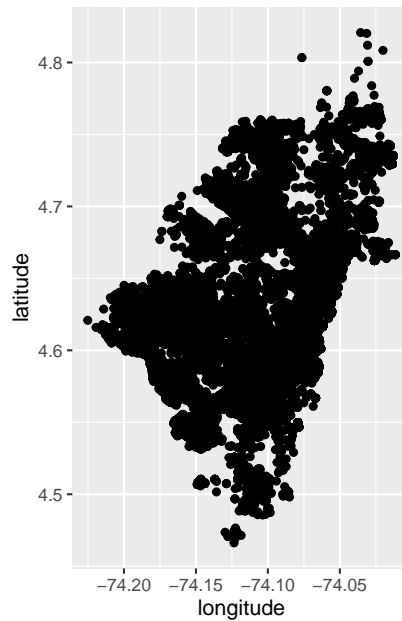
# Nociones básicas de vizualización

Hay varias maneras de vizualizar la infromación espacial en **R**. Sin embargo, partiremos de la manera mas “rápida” y esta es a través de la utilización de `ggplot`.

### 4.1 Vizualizar lo más elemental: una base de datos con coordenadas

Cuando tenemos una base de puntos, podemos vizualizarla con `ggplot2`, pero esta vizualización no tiene referencias como un “mapa base”. Con el comando `coord_equal()` podemos hacer que los vertices de las coordeandas sean siempre proporcionales entre sí:

```
a01_homicidios %>% ggplot(aes(longitude, latitude))+  
  geom_point()+coord_equal()
```



A pesar de que parece un mecanismo simple hay incluso otros comandos dentro de ggplot2 muy útiles de para vizualizar incuso densidades como `geom_bin2d()` y `geom_density2d()`

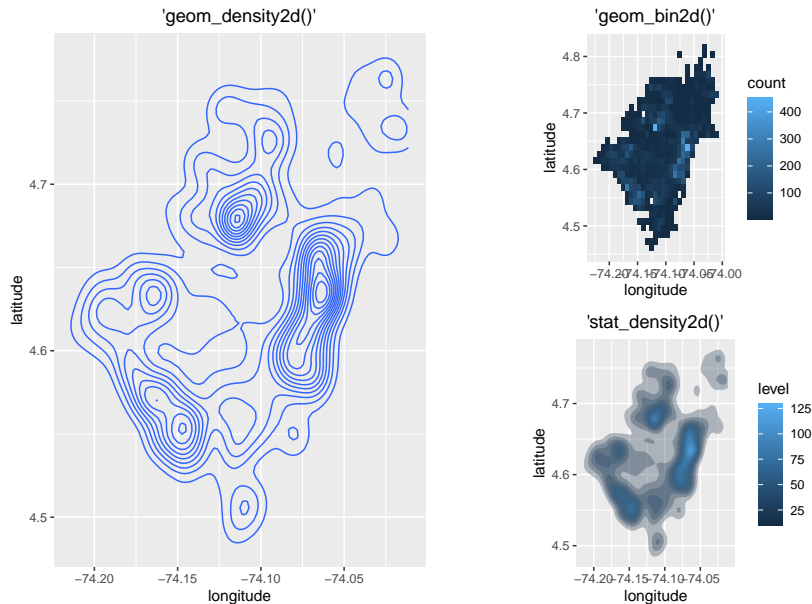
```
ggpubr::ggarrange(
  # Lineas de nivel de densidad
  a01_homicidios %>% ggplot(aes(longitude, latitude))+
    geom_density2d()+coord_equal()+labs(title = "'geom_density2d()'")+
    theme(plot.title = element_text(hjust = 0.5)),

  ggpubr::ggarrange(

    # Densidades simples
    a01_homicidios %>% ggplot(aes(longitude, latitude))+
      geom_bin2d()+coord_equal()+labs(title = "'geom_bin2d()'")+
      theme(plot.title = element_text(hjust = 0.5)),

    # Densidades lineales
    a01_homicidios %>% ggplot(aes(longitude, latitude))+
      stat_density2d(aes(fill = ..level..),
        geom = "polygon", alpha = 0.3)+coord_equal()+
      labs(title = "'stat_density2d()'")+
      theme(plot.title = element_text(hjust = 0.5)),
    nrow = 2),
```

```
nrow = 1)
```



## 4.2 EXTRA: Agregando un mapa base con ‘ggmap()’

Aunque es posible realizar mapas simplemente modificando algunas características con ggplot2 de base, hay una librería `ggmap()` que es una librería que nos puede ayudar a importar mapas base. Esto no es por lo general necesario, pero en algunos casos puede ser útil para que los lectores del mapa se guíen más, o para darle un acabado más estético a los mapas.

### LO NO TAN CHEVRE:

Lo no tan chevre, es que para el uso de esta librería hay que habilitar el API de Google Maps. Los pasos para habilitar el API son sencillos, pero es necesario registrar una tarjeta de crédito en GCP (Google Cloud Platform) para que la aplicación no otorgue una *API Key*. No obstante, a pesar de que tenemos que registrar una tarjeta de crédito, dado el bajo consumo que (por lo general) nuestros análisis requieran, no generará cobros. Más información de esto puede verse con el comando `help("register_google")`.

Para agregar una base a estos mapas simples creados desde ggplot, usaremos la librería `ggmap()`.

```

library(ggmap)
# bogota_mapa_a <- get_map(c("lon" = -74.10936, "lat" = 4.628712), zoom = 11,
#                           source = "stamen", maptype = "watercolor")
# bogota_mapa_b <- get_map(c("lon" = -74.10936, "lat" = 4.628712), zoom = 11,
#                           source = "cloudmade", maptype = 53428)
#
#
#
# ggpubr::ggarrange(
#   ggmap(bogota_mapa_a,
#         base_layer = a01_homicidios %>%
#           filter(year == 2019) %>%
#             ggplot(aes(longitude, latitude)))+
#     geom_density2d(color = "red", alpha = 0.7)+
#     labs(title = "Densidad de homicidios Bogotá-2019",
#          subtitle = "stamen:watercolor",
#          caption = "Fuente: Policía Metropolitana de Bogotá")+
#     theme(text = element_text(family = "serif"),
#           plot.title = element_text(hjust = 0.5),
#           plot.subtitle = element_text(hjust = 0.5),
#           axis.text = element_blank(),
#           axis.title = element_blank()),
#   ggmap(bogota_mapa_b,
#         base_layer = a01_homicidios %>%
#           filter(year == 2019) %>%
#             ggplot(aes(longitude, latitude)))+
#     geom_density2d(color = "red", alpha = 0.7)+
#     labs(title = "Densidad de homicidios Bogotá-2019",
#          subtitle = "stamen:toner",
#          caption = "Fuente: Policía Metropolitana de Bogotá")+
#     theme(text = element_text(family = "serif"),
#           plot.title = element_text(hjust = 0.5),
#           plot.subtitle = element_text(hjust = 0.5),
#           axis.text = element_blank(),
#           axis.title = element_blank()),
#   nrow = 1)

```

Al igual que casi todo el resto de recursos de R, hay varias *cheat-cheats* y guías en el internet que podemos consultar para recordar un poco mejor que parámetros podemos emplear en `source` y `maptype`.