

# 01-conocer.Rmd

Herramientas y Datos para conocer el Territorio

Angie Scetta

Octubre 2020



# Índice general

<b>INTRODUCCIÓN</b>	<b>5</b>
Indice . . . . .	5
<b>1 PRESENTACIÓN</b>	<b>7</b>
1.1 Interfaz Gráfica de RStudio . . . . .	7
1.2 Formato RMarkdown . . . . .	9
1.3 Nuestro primer dataset . . . . .	10
<b>2 MANIPULACIÓN DE DATOS</b>	<b>17</b>
2.1 Filtrar registros que cumplan condiciones . . . . .	18
2.2 Seleccionar columnas de interés . . . . .	26
2.3 Modificar o agregar columnas . . . . .	30
2.4 Ordenar registros . . . . .	36
2.5 Renombrar columnas . . . . .	40
2.6 Resumir y agrupar datos . . . . .	42
2.7 Concatenar funciones (%>%) . . . . .	44
2.8 Transformar la estructura de los datos . . . . .	46
<b>3 ANÁLISIS Y VISUALIZACIÓN DE DATOS</b>	<b>51</b>
3.1 Distribución de una variable continua . . . . .	52
3.2 Distribución de valores continuos asociados a una variable categórica: Gráfico de Cajas . . . . .	63
3.3 Relación entre variables numéricas: Gráfico de Dispersión . . . . .	69
3.4 Relación entre variables categóricas: Gráfico de Matriz . . . . .	75
3.5 Relación entre variable numérica y categórica: Gráfico de Barras	79

<b>4 INFORMACIÓN GEOGRÁFICA Y MAPAS</b>	<b>89</b>
4.1 Analizar datos espaciales . . . . .	90
4.2 Cruzar datos tradicionales y espaciales . . . . .	99
4.3 Cruzar datos espaciales . . . . .	110
4.4 Agregar mapa base . . . . .	121

# INTRODUCCIÓN

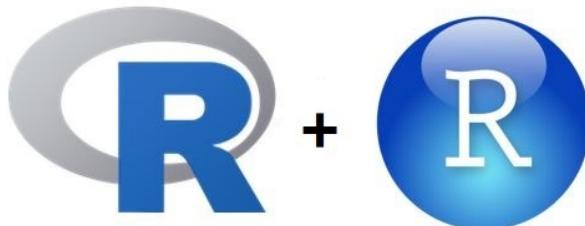
¡Hola a todos!

## ¡Bienvenidos al mundo de los datos!

En la actualidad las Ciudades producen millones de datos por segundo y hoy tenemos la oportunidad de aprovecharlos desde nuestra profesión para poder trabajar y tomar decisiones en base a evidencia real.

El objetivo del manual es presentarle a los profesionales interesados en analizar diversas variables territoriales (sociodemográficas, mercado inmobiliario, actividad económica, transporte, movilidad, salud, educación, etc), la importancia de conocer herramientas analíticas que les permitan manipular y extraer conocimiento de grandes volúmenes de datos para tomar mejores decisiones a la hora de diagnosticar, comprender e intervenir el territorio.

Concretamente, el manual introducirá los conocimientos básicos del lenguaje de programación R y del software libre RStudio, y su potencialidad para el análisis de datos urbanos.



A lo largo de cada módulo se aplicará la herramienta a datos reales, con el fin de analizar, describir, interpretar, visualizar y extraer conocimiento de datos existentes.

## Índice

### Módulo 1: INTRODUCCIÓN BIG DATA Y CIUDAD

¿Qué es la Ciencia de Datos? ¿Cómo utilizar grandes volúmenes de datos para analizar dinámicas urbanas? ¿Qué datos existen? ¿Qué información contienen las bases de datos? ¿Cómo se estructuran?

Introducción a programación en lenguaje R y el software RStudio.

Para esta clase será necesario tener previamente instalado el lenguaje de programación R (<https://cloud.r-project.org/>), y la interfaz gráfica RStudio Desktop (<https://rstudio.com/products/rstudio/download/>).

En el caso de no poder instalar correctamente el programa, se recomienda utilizar RStudio Cloud (<https://rstudio.cloud/>).

## Módulo 2: MANIPULACIÓN DE DATOS

Manipular y transformar los datos: Técnicas de *data wrangling*.

¿Cómo extraer información útil de los datos? ¿Cómo encontrar información específica?

Análisis exploratorio de datos. Las funciones de *tidyverse*: Seleccionar, Filtrar, Ordenar, Modificar, Resumir, Agrupar y Renombrar los datos.

## Módulo 3: ANÁLISIS Y VISUALIZACIÓN DE DATOS

Análisis y Visualización de Información.

¿Por qué representar gráficamente los datos? ¿Qué tipo de visualización conviene usar?

Extraer conocimiento de los datos a partir de la correcta visualización de los mismos. Comunicar resultados desarrollando gráficos con *ggplot*.

## Módulo 4: INFORMACIÓN GEOGRÁFICA Y MAPAS

Análisis y Visualización de Información Geográfica.

¿Qué son los Sistemas de Información Geográfica (SIG)? ¿Por qué mapear datos? ¿Dónde se ubican los datos? ¿Cómo se distribuyen en el territorio? ¿Presentan algún patrón espacial?

Análisis exploratorio de datos espaciales. Adquirir conocimiento de las herramientas básicas del paquete *sf* para manipular información geográfica teniendo en cuenta formatos, sistemas de coordenadas y proyecciones. Generación de mapas temáticos con *ggplot*.

# Capítulo 1

## PRESENTACIÓN

### ¿Qué es R y RStudio?

RStudio es una **interfaz libre y gratuita** que nos permite **explotar todo el potencial que tiene el lenguaje de programación R**.

R es un lenguaje que ofrece una gran variedad de funciones para realizar cálculos estadísticos y generar diversos gráficos a partir de los datos. Sin embargo, el gran potencial está en que, al ser libre y colaborativo, **constantemente los usuarios están actualizando y ampliando la cantidad de funciones que presenta**. Hoy en día podemos realizar desde operaciones básicas sobre los datos hasta aplicar algoritmos de inteligencia artificial.

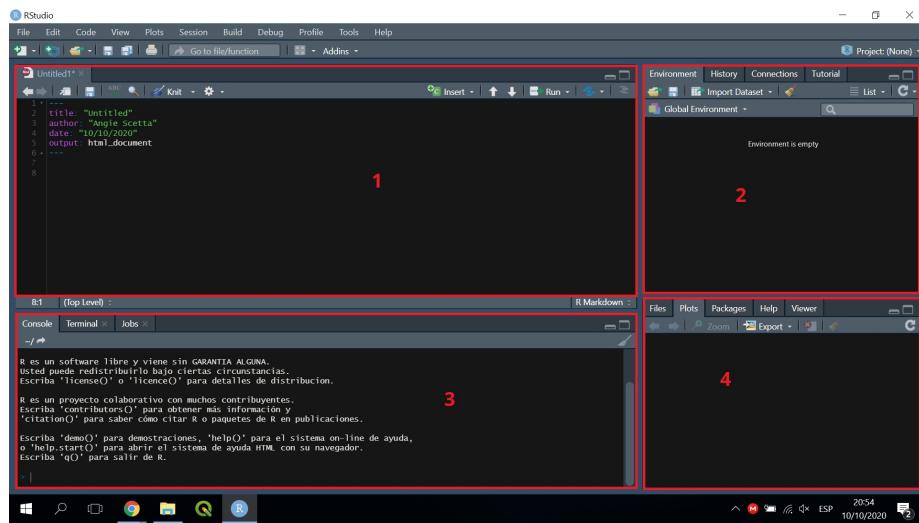
A su vez, dentro de RStudio, hay diferentes formatos de archivos (RMarkdown, RScript, RNotebook, etc) y su elección depende del objetivo que tengamos. En nuestro caso, a lo largo del manual trabajaremos con el formato **RMarkdown**, un tipo de documento de RStudio que integra texto con código de R y nos permite generar informes a partir de los datos.

Empecemos de a poco:

Lo primero que tenemos que hacer es crear un **nuevo proyecto y un nuevo RMarkdown**. Los pasos a seguir pueden encontrarlos en el siguiente tutorial: <https://rpubs.com/angiescetta/conociendo-R>

### 1.1 Interfaz Gráfica de RStudio

Antes de seguir, analicemos un poco la **Interfaz Gráfica de RStudio**:



Tal como se ve en la imagen, podríamos dividir la interfaz en 4 partes/ventanas:

### 1. Panel de Edición

Este panel es en el que vamos a estar creando y modificando nuestro RMarkdown. Aquí también podría haber otro formato de archivo R, como por ejemplo R Script o R Notebook.

### 2. Entorno de Variables

En esta ventana iremos viendo todos los datos que hayamos cargado. Desde aquí también podremos importar o eliminar datos.

Desde la pestaña Historial podremos consultar el historial de comandos y funciones que fuimos utilizando en el Proyecto.

### 3. Consola

En la ventana inferior izquierda irá apareciendo todo lo que ejecutemos tanto desde el Panel de Edición como desde el Entorno de Variables, pero también podemos escribir líneas de código que queremos que se ejecuten y no queremos dejarlas escritas en el RMarkdown.

### 4. Panel de Utilidades

En la ventana inferior derecha se pueden ver varias cosas:

- Files: El Directorio donde estamos trabajando.

- b. Plots: Las visualizaciones/gráficos que se van generando.
- c. Packages: Los paquetes de R disponibles.
- d. Help: Una sección de ayuda donde podemos consultar información de las funciones.
- e. Viewer: Un visor HTML para ver los gráficos interactivos o animados que hayamos hecho.

## 1.2 Formato RMarkdown

### ¿Para que sirve el formato RMawkdown?

Este formato sirve para manipular datos y armar informes listos para presentar. Hay 2 formas de escribir en un RMarkdown:

1. **Texto** como el que estoy escribiendo ahora.

El RMarkdown tiene una sintaxis específica para poder dar formato al texto del informe final, por ejemplo si escribimos así:

```

30 * # Clase 1 del curso BIG DATA URBANA -> Encabezado 1
31
32 *## Clase 1 del curso BIG DATA URBANA -> Encabezado 2
33
34 *### Clase 1 del curso BIG DATA URBANA -> Encabezado 3
35
36 *#### Clase 1 del curso BIG DATA URBANA -> Encabezado 4
37
38 *##### Clase 1 del curso BIG DATA URBANA -> Encabezado 5
39
40 Clase 1 del curso BIG DATA URBANA -> Normal
41
42 *Clase 1 del curso BIG DATA URBANA* -> Cursiva
43
44 **Clase 1 del curso BIG DATA URBANA** -> Negrita
45
46 1. Clase 1 del curso BIG DATA URBANA -> Enumeración de ítems
47
48 * Clase 1 del curso BIG DATA URBANA -> Punteo de ítems
49
50 + Clase 1 del curso BIG DATA URBANA -> Subpunteo de ítems
51

```

Obtenemos los siguientes resultados:

**Clase 1 del curso BIG DATA URBANA -> Encabezado 1**

**Clase 1 del curso BIG DATA URBANA -> Encabezado 2**

Clase 1 del curso BIG DATA URBANA -> Encabezado 3

Clase 1 del curso BIG DATA URBANA -> Encabezado 4

Clase 1 del curso BIG DATA URBANA -> Encabezado 5

Clase 1 del curso BIG DATA URBANA -> Normal

*Clase 1 del curso BIG DATA URBANA -> Cursiva*

**Clase 1 del curso BIG DATA URBANA -> Negrita**

1. Clase 1 del curso BIG DATA URBANA -> Enumeración de ítems

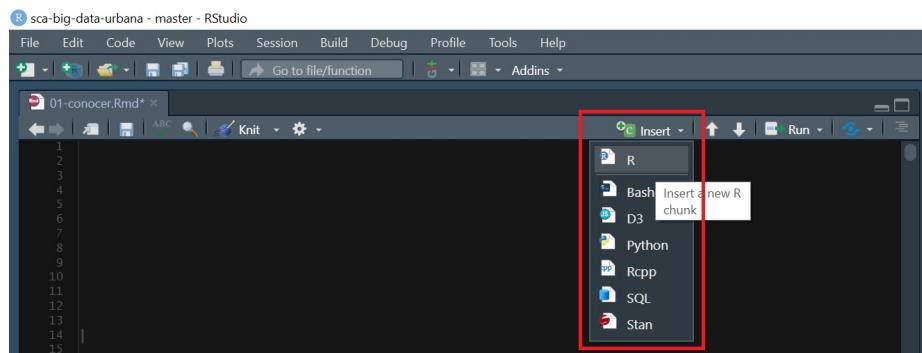
• Clase 1 del curso BIG DATA URBANA -> Punteo de ítems

◦ Clase 1 del curso BIG DATA URBANA -> Subpunteo de ítems

2. **Bloques de código** (o “chunks”) donde insertaremos nuestras líneas de código con el objetivo de manipular (analizar, modificar, visualizar) los datos. Esto es un chunk:



Y se inserta haciendo click en **Insert/R** o con el siguiente atajo en el teclado: **Ctrl + Shift + I**



### 1.3 Nuestro primer dataset

En los primeros capítulos del manual vamos a trabajar con los datos de Properati, un portal web de compra, venta y alquiler de inmuebles en toda América Latina. Estos datos son públicos y pueden encontrarlos en <https://properati.com.ar/data>.

En este caso, para facilitar la manipulación de la información, usaremos un set de datos (en formato csv) previamente procesado que contiene datos de propiedades publicadas en AMBA en Junio y Julio del 2020. Pueden descargarlo de <https://data.world/angie-scetta/amba-properati>.

*Recomendación: Al descargarlo, moverlo de la carpeta “Descargas” a una nueva carpeta llamada “data” dentro de la carpeta del Proyecto donde estén trabajando.*

Ahora si, manos a la obra! Para cargar el dataset pueden copiar la siguiente línea de código y pegarla dentro de un chunk:

```
datos_amba <- read.csv("data/amba_properati.csv")
```

Para entender la lógica detrás del chunk anterior pueden revisar este link: <https://rpubs.com/angiescetta/importar-dataset>

**Ahora conocemos nuestro dataset:** Veamos como se estructura (cuantas filas y columnas tiene) y que información trae...

Para esto empezaremos utilizando `dim()`:

```
dim(datos_amba)
```

```
## [1] 14929    13
```

Podemos ver que tenemos 14.929 registros/filas y 13 columnas. También podríamos ver esto por separado de la siguiente forma:

```
ncol(datos_amba)
```

```
## [1] 13
```

```
nrow(datos_amba)
```

```
## [1] 14929
```

Pero ¿Qué información contienen esas 13 columnas?

Esto podemos verlo con `names()`:

```
names(datos_amba)
```

```
## [1] "created_on"      "provincia"       "partido"        "rooms"
## [5] "surface_total"   "surface_covered" "price"          "currency"
## [9] "title"           "property_type"  "operation_type" "lat"
## [13] "lon"
```

Bien, las columnas tienen fecha de publicación de la propiedad, provincia y partido donde se ubica, cantidad de ambientes, superficie total y superficie cubierta, precio publicado, tipo de moneda (ARS o USD), el título que el usuario escribió al publicar su propiedad, el tipo de propiedad, el tipo de operación, y finalmente la ubicación del inmueble con sus coordenadas: latitud y longitud.

Parece que cada fila/registro de la base corresponde a una propiedad publicada, pero veamos una pequeña muestra de la data con `head()` para estar seguros:

```
head(datos_amba)
```

	created_on	provincia	partido	rooms	surface_total	surface_covered	price
## 1	202006	CABA	Comuna 7	1	40		37 22500
## 2	202006	CABA	Comuna 13	1	30		30 18000
## 3	202006	CABA	Comuna 13	1	31		29 17900

```

## 4    202006      CABA      Comuna 1      1      35      35 42000
## 5    202006      GBA   Vicente López      1      36      27 19000
## 6    202006      GBA   La Matanza      2      24      24 12000
## currency
## 1      ARS
## 2      ARS
## 3      ARS
## 4      ARS
## 5      ARS
## 6      ARS
##
## title
## 1          Departamento - Flores
## 2          Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3          Departamento - Belgrano
## 4          Monoambiente con cochera. Zencity. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6          PH - Lomas Del Mirador
## property_type operation_type      lat      lon
## 1 Departamento      Alquiler -34.61917 -58.46222
## 2 Departamento      Alquiler -34.55460 -58.46652
## 3 Departamento      Alquiler -34.56318 -58.46461
## 4 Departamento      Alquiler -34.61836 -58.36090
## 5 Departamento      Alquiler -34.53344 -58.49345
## 6          PH      Alquiler -34.66253 -58.52914

```

Y un resumen estadístico de la información:

```
summary(datos_amba)
```

```

## created_on      provincia      partido      rooms      surface_total
## Min. :202006  CABA:8896  Comuna 14:2001  Min. : 1.000  Min. : 10
## 1st Qu.:202006  GBA :6033  Tigre :1222    1st Qu.: 2.000  1st Qu.: 51
## Median :202006                      Comuna 13:1193  Median : 3.000  Median : 75
## Mean   :202006                      Comuna 15:1077  Mean   : 3.057  Mean   : 117
## 3rd Qu.:202007                      Comuna 2 : 853   3rd Qu.: 4.000  3rd Qu.: 130
## Max.   :202007                      Comuna 1 : 765   Max.   :10.000  Max.   :5000
## 
## (Other) :7818
## surface_covered      price      currency
## Min. : 10.00  Min. : 10000  ARS: 3251
## 1st Qu.: 46.00  1st Qu.: 72000  USD:11678
## Median : 66.00  Median : 140000
## Mean   : 91.98  Mean   : 206342
## 3rd Qu.:108.00  3rd Qu.: 250000
## Max.   :882.00  Max.   :3000000
## 
```

```

##                                     title
## Departamento de 2 ambientes en Venta en Villa crespo: 182
## Departamento de 2 ambientes en Venta en Almagro      : 124
## Departamento de 2 ambientes en Venta en Palermo       : 124
## Departamento de 3 ambientes en Venta en Villa crespo: 121
## Departamento de 3 ambientes en Venta en Almagro      : 108
## Departamento de 3 ambientes en Venta en Palermo       : 108
## (Other)                                              :14162
##          property_type   operation_type      lat        lon
## Casa         : 2297   Alquiler: 3251   Min.   :-35.12   Min.   :-59.04
## Departamento:11344   Venta    :11678   1st Qu.:-34.61   1st Qu.:-58.53
## PH           : 1288                               Median :-34.59   Median :-58.45
##                                         Mean   :-34.59   Mean   :-58.47
##                                         3rd Qu.:-34.55   3rd Qu.:-58.41
##                                         Max.   :-34.26   Max.   :-57.83
##

```

A priori, en este resumen podemos entender varias cosas de la data. Por ejemplo:

- Hay registros de propiedades publicadas entre **Junio (Min) y Julio (Max) 2020**.
- La **mayoría de las publicaciones son en CABA** (8.896 vs 6.033).
- El **Partido/Comuna que más aparece es la Comuna 14** (Barrio de Palermo), seguido por Tigre.
- De las 3 tipologías de propiedades, **lo que más hay es Departamentos**, seguido por Casas y por último PHs.
- Hay **más propiedades en Venta que en Alquiler** (11.678 vs 3.251).

Por último, investiguemos como es la **estructura de la data**, es decir que tipo de información tiene cada campo. Si bien algo ya nos imaginamos gracias al `summary()`, usemos `str()` para revisarlo:

```
str(datos_amba)
```

```

## 'data.frame': 14929 obs. of  13 variables:
## $ created_on     : int  202006 202006 202006 202006 202006 202006 202006 202006 202006 ...
## $ provincia      : Factor w/ 2 levels "CABA","GBA": 1 1 1 1 2 2 2 1 1 1 ...
## $ partido        : Factor w/ 50 levels "Almirante Brown",...: 18 10 10 6 50 31 27 10 10 6 ...
## $ rooms          : int  1 1 1 1 1 2 2 2 2 ...
## $ surface_total   : int  40 30 31 35 36 24 40 60 53 39 ...
## $ surface_covered: int  37 30 29 35 27 24 40 50 44 34 ...
## $ price          : int  22500 18000 17900 42000 19000 12000 15000 32000 26000 14000 ...

```

```
## $ currency      : Factor w/ 2 levels "ARS","USD": 1 1 1 1 1 1 1 1 1 ...
## $ title         : Factor w/ 10634 levels "- Casa Venta Tipo Chalet 5 Amb con Cochera ...
## $ property_type : Factor w/ 3 levels "Casa","Departamento",...: 2 2 2 2 2 3 3 2 2 ...
## $ operation_type: Factor w/ 2 levels "Alquiler","Venta": 1 1 1 1 1 1 1 1 1 ...
## $ lat            : num -34.6 -34.6 -34.6 -34.6 -34.5 ...
## $ lon            : num -58.5 -58.5 -58.5 -58.4 -58.5 ...
```

Existen varios tipos de datos, pero en nuestro set nos encontramos con 3: integer (int), numeric (num) y Factor.

Ahora bien, ¿Qué significa eso?

- **Integer** son números enteros, es decir que el campo solo admite números sin decimales como por ejemplo cantidad de habitaciones de una propiedad.
- **Numeric** son números con decimales, como por ejemplo, latitud y longitud.
- **Factor** son categorías, por ejemplo Barrios, Partidos, Tipos de propiedades, etc.

Además del tipo de dato, con `str()` también podemos ver la cantidad de niveles que tienen las variables de tipo Factor. Por ejemplo, se observa que la columna provincia solo tiene 2 categorías posibles (CABA o AMBA), en cambio la columna partido tiene 50.

### Bonus Track

¿Cómo hacemos si queremos ver la estructura o un resumen estadístico de una sola de las columnas del dataset? Para esto utilizamos el símbolo `$` de la siguiente forma:

```
summary(datos_amba$property_type)

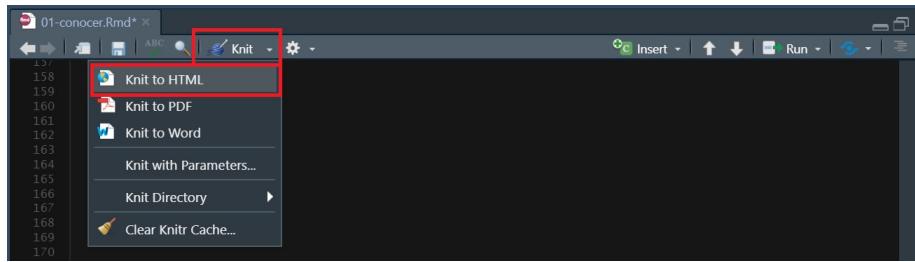
##          Casa Departamento        PH
##          2297       11344       1288

str(datos_amba$property_type)

##  Factor w/ 3 levels "Casa","Departamento",...: 2 2 2 2 2 3 3 2 2 2 ...
```

### Generemos nuestro primer HTML

Por último, generemos nuestro primer reporte HTML para poder ver todos los resultados en un único informe. Para esto debemos hacer click en **Knit / Knit to HTML**:



### Próximos Pasos

Acá concluye la primer clase, pero los invito a que repliquen lo realizado con algún otro dataset que les interese. Pueden descargar datos de diversos portales abiertos como por ejemplo:

- Portal de Datos Abiertos de Argentina: <https://datos.gob.ar/>
- Portal de Datos Abiertos de CABA: <https://data.buenosaires.gob.ar/>
- Portal de Datos Abiertos de PBA: <https://catalogo.datos.gba.gob.ar/>
- O de cualquier otro portal de datos que conozcan o encuentren!



## Capítulo 2

# MANIPULACIÓN DE DATOS

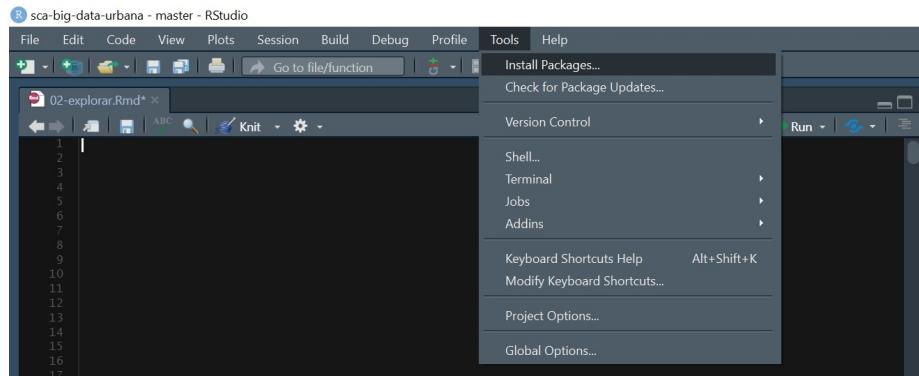
Ahora que ya sabemos abrir un dataset y conocer que información tiene, vamos a aprender a manipular, limpiar, normalizar y transformar los datos, o lo que se conoce como *data wrangling*. Para esto vamos a trabajar con uno de los paquetes más usados y más útiles de R que se llama **tidyverse**.

Pero, ¿Qué es un paquete?

Cuando instalamos R ya viene con múltiples funciones básicas para manipular datos, sin embargo el potencial de la herramienta surge con la posibilidad de incorporar constantemente nuevas funciones que nos permitan realizar nuevas tareas o mejorar el resultado de las ya existentes.

Estos grupos de funciones son a los que llamamos paquetes o packages y para poder utilizarlos es necesario **instalarlos por única vez en la computadora**, y luego **activarlos cada vez que vayamos a usarlos**.

Comencemos instalándolo. Esto podemos hacerlo manualmente en Tools/Install packages:



O directamente escribiendo `install.packages()` adentro de un chunk:

```
#install.packages("tidyverse")
```

Una vez que instalamos el paquete, no vamos a tener que volver a hacerlo. Solamente vamos a tener que “activarlo” cada vez que queramos usarlo. Esta activación se hace con `library()` así:

```
library(tidyverse)
```

Ahora volvamos a cargar nuestro dataset (el mismo de la clase anterior):

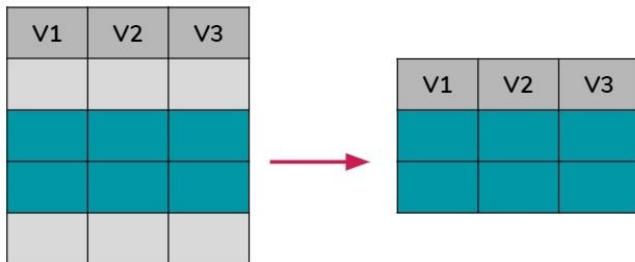
```
datos_amba <- read.csv("data/amba_properati.csv")
```

A continuación veremos como el paquete `tidyverse` nos va a permitir manipular nuestros datos a partir de las funciones: **filtrar**, **modificar**, **seleccionar**, **ordenar**, **renombrar**, **resumir** y **agrupar**.

Aprender a utilizar todas estas funciones es muy importante ya que la **comprensión, transformación y limpieza de los datos** es la etapa que más tiempo nos llevará a la hora de encarar cualquier proyecto de Ciencia de Datos.

## 2.1 Filtrar registros que cumplan condiciones

Como su nombre lo indica, esta función hace referencia a realizar un filtro determinado sobre los registros/filas de toda la base de datos, es decir, quedarnos solo con las filas que cumplan cierta condición establecida. Gráficamente se vería como algo así:



Esto nos será muy útil si por algún motivo queremos dejar de lado registros y utilizar solo una parte de la base. Por ejemplo, en el caso de nuestro dataset, que ya vimos que incluye datos de AMBA y CABA, podríamos filtrar la data y quedarnos solo con los registros ubicados en CABA:

```
filtro <- filter(datos_amba, provincia=="CABA")

head(filtro)

##   created_on provincia partido rooms surface_total surface_covered price
## 1 202006      CABA Comuna 7     1          40            37 22500
## 2 202006      CABA Comuna 13    1          30            30 18000
## 3 202006      CABA Comuna 13    1          31            29 17900
## 4 202006      CABA Comuna 1     1          35            35 42000
## 5 202006      CABA Comuna 13    2          60            50 32000
## 6 202006      CABA Comuna 13    2          53            44 26000
##   currency                                              title
## 1 ARS          Departamento - Flores
## 2 ARS          Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3 ARS          Departamento - Belgrano
## 4 ARS          Monoambiente con cochera. Zencity. Puerto Madero
## 5 ARS Depto 2AMB c/ Suite y Toilette - Deheza 1600 - Incluye Cochera
## 6 ARS          Unico 2 ambientes A Estrenar! Posesion inmediata! - Nuñez
##   property_type operation_type      lat      lon
## 1 Departamento    Alquiler -34.61917 -58.46222
## 2 Departamento    Alquiler -34.55460 -58.46652
## 3 Departamento    Alquiler -34.56318 -58.46461
## 4 Departamento    Alquiler -34.61836 -58.36090
## 5 Departamento    Alquiler -34.53795 -58.46671
## 6 Departamento    Alquiler -34.54756 -58.47037
```

O los del mes de Julio 2020:

```
filtro <- filter(datos_amba, created_on==202007)

head(filtro)
```

```

##   created_on provincia partido rooms surface_total surface_covered price
## 1    202007      GBA     Tigre     2        108       69 20000
## 2    202007      GBA     Tigre     2         57       45 27500
## 3    202007     CABA Comuna 14     1        45       40 23000
## 4    202007     CABA Comuna 14     2        48       48 30000
## 5    202007     CABA Comuna 14     2        42       37 26000
## 6    202007     CABA Comuna 15     2        41       38 21000
##   currency
## 1      ARS DEPARTAMENTO EN ALQUILER EN NORDELTA 2 AMB CON TERRAZA Y JACUZZI
## 2      ARS      2 ambientes con cochera y baulera - Vista al Lago Central
## 3      ARS          Departamento en Alquiler con Balcón y Amenities
## 4      ARS      Botánico - 2 ambientes amueblado con vista a Boulevard Cerviño
## 5      ARS          Departamento - Palermo
## 6      ARS          Departamento de 2 ambientes en Alquiler en Palermo
##   property_type operation_type      lat      lon
## 1 Departamento      Alquiler -34.39437 -58.64925
## 2 Departamento      Alquiler -34.40910 -58.63129
## 3 Departamento      Alquiler -34.59274 -58.42749
## 4 Departamento      Alquiler -34.57914 -58.41289
## 5 Departamento      Alquiler -34.59571 -58.41587
## 6 Departamento      Alquiler -34.59856 -58.43026

```

Nótese que para filtrar bajo la condición de “igual” utilicé “==”, pero también podría haber utilizado otras condiciones como:

- **A==B** -> A igual a B
- **A!=B** -> A diferente a B
- **A<B** -> A menor a B
- **A<=B** -> A menor o igual a B
- **A>B** -> A mayor a B
- **A>=B** -> A mayor o igual a B
- **is.na(A)** -> A tiene valor nulo (NA)
- **!is.na(A)**-> A no tiene valor nulo (NA)
- **A%in%B** -> A incluye el valor B
- **!(A%in%B)**-> A no incluye el valor de B

Entonces, por ejemplo si quiero quedarme solo con las propiedades que tienen una superficie cubierta mayor o igual a 75m<sup>2</sup> debería escribirlo así:

```

filtrado <- filter(datos_amba, surface_covered>=75)

head(filtrado)

```

```

##   created_on provincia partido rooms surface_total surface_covered price

```

```

## 1 202006 GBA Ituzaingó 3 75 75 19000
## 2 202006 GBA Tigre 3 95 80 42000
## 3 202006 GBA Tigre 3 98 90 42000
## 4 202006 CABA Comuna 13 3 109 94 90000
## 5 202006 CABA Comuna 14 3 152 152 240000
## 6 202006 CABA Comuna 14 3 90 85 80000
## currency
## 1 ARS
## 2 ARS
## 3 ARS
## 4 ARS
## 5 ARS
## 6 ARS
##
## Americana a 6 cuadras de la estacion de S.A. de P
## 3 AMBIENTES EN EDIFICIO MIRADORES DE LA BAHIA CON COCHERA
## 3 ¡EN EXCLUSIVA! Departamento en alquiler de 3 ambientes Miradores de La Bahía, Bahía de Norde
## 4 Alquiler Depto 3 Amb a Estrenar Nuñez C/Coc
## 5 ALQUILER en TORRES DE GELLY 152 m2 con 2 dorm dep y 1 coch ALTO vista Rio SIN mue
## 6 ALQUILER 3 AMB CON DEPENDENCIA! - Y COCHERA TORRE AMENITI
## property_type operation_type lat lon
## 1 Casa Alquiler -34.65957 -58.69712
## 2 Departamento Alquiler -34.39642 -58.64665
## 3 Departamento Alquiler -34.39512 -58.64665
## 4 Departamento Alquiler -34.54521 -58.46286
## 5 Departamento Alquiler -34.57760 -58.40499
## 6 Departamento Alquiler -34.59353 -58.41462

```

O si quiero eliminar los registros que corresponden a departamentos debería hacer el siguiente chunk:

```

filtro <- filter(datos_amba, property_type!="Departamento")

head(filtro)

```

```

## created_on provincia partido rooms surface_total surface_covered
## 1 202006 GBA La Matanza 2 24 24
## 2 202006 GBA General San Martín 2 40 40
## 3 202006 GBA Ituzaingó 3 75 75
## 4 202006 CABA Comuna 14 4 125 105
## 5 202006 GBA Merlo 4 180 140
## 6 202006 GBA Pilar 4 267 267
## price currency title
## 1 12000 ARS PH - Lomas Del Mirador
## 2 15000 ARS PH - Chilavert

```

```

## 3 19000      ARS Americana a 6 cuadras de la estacion de S.A. de Padua
## 4 85000      ARS             PH en alquiler 4 ambientes - Las Cañitas
## 5 27000      ARS             Casa en Padua 3 dormitorios en alquiler
## 6 50000      ARS             Casa en venta o alquiler en La Peregrina
##   property_type operation_type      lat      lon
## 1           PH     Alquiler -34.66253 -58.52914
## 2           PH     Alquiler -34.54301 -58.57573
## 3           Casa    Alquiler -34.65957 -58.69712
## 4           PH     Alquiler -34.57207 -58.43249
## 5           Casa    Alquiler -34.66154 -58.71565
## 6           Casa    Alquiler -34.47091 -58.82270

```

Si queremos filtrar todas las propiedades ubicadas en 3 partidos diferentes como por ejemplo La Plata, General San Martín y La Matanza debemos utilizar %in% de la siguiente forma:

```

filtro <- filter(datos_amba, partido %in% c("La Plata", "General San Martín", "La Matanza"))

head(filtro)

##   created_on provincia          partido rooms surface_total surface_covered
## 1 202006       GBA        La Matanza     2      24            24
## 2 202006       GBA General San Martín     2      40            40
## 3 202006       GBA        La Plata      2      50            50
## 4 202006       GBA        La Plata      2      35            35
## 5 202006       GBA General San Martín     2      56            48
## 6 202006       GBA        La Plata      3      40            40
##   price currency                                         title
## 1 12000      ARS             PH - Lomas Del Mirador
## 2 15000      ARS             PH - Chilavert
## 3 18000      ARS Alquiler, depto de un domitorio frente al parque San Martin
## 4 13500      ARS                                         Departamento - La Plata
## 5 14000      ARS                                         Departamento céntrico - Villa Ballester
## 6 10000      ARS                                         Departamento - José Hernández
##   property_type operation_type      lat      lon
## 1           PH     Alquiler -34.66253 -58.52914
## 2           PH     Alquiler -34.54301 -58.57573
## 3 Departamento    Alquiler -34.93415 -57.96605
## 4 Departamento    Alquiler -34.90645 -57.97335
## 5 Departamento    Alquiler -34.55038 -58.55512
## 6 Departamento    Alquiler -34.89768 -58.02787

```

En cambio, si queremos filtrar todas las propiedades que no estén ubicadas en 3 partidos diferentes como por ejemplo La Plata, General San Martín y La Matanza debemos utilizar ! + %in% de la siguiente forma:

```

filtro <- filter(datos_amba, !(partido %in% c("La Plata", "General San Martín", "La Matanza")))

head(filtro)

##   created_on provincia      partido rooms surface_total surface_covered price
## 1 202006      CABA       Comuna 7     1           40             37 22500
## 2 202006      CABA       Comuna 13    1           30             30 18000
## 3 202006      CABA       Comuna 13    1           31             29 17900
## 4 202006      CABA       Comuna 1     1           35             35 42000
## 5 202006      GBA Vicente López 1           36             27 19000
## 6 202006      CABA       Comuna 13    2           60             50 32000

##   currency
## 1     ARS
## 2     ARS
## 3     ARS
## 4     ARS
## 5     ARS
## 6     ARS

##   title
## 1 Departamento - Flores
## 2 Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3 Departamento - Belgrano
## 4 Monoambiente con cochera. Zencyt. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6 Depto 2AMB c/ Suite y Toilette - Deheza 1600 - Incluye Cochera

##   property_type operation_type      lat      lon
## 1 Departamento     Alquiler -34.61917 -58.46222
## 2 Departamento     Alquiler -34.55460 -58.46652
## 3 Departamento     Alquiler -34.56318 -58.46461
## 4 Departamento     Alquiler -34.61836 -58.36090
## 5 Departamento     Alquiler -34.53344 -58.49345
## 6 Departamento     Alquiler -34.53795 -58.46671

```

Pero esto no es todo, ¿Cómo hago si quiero filtrar por **2 o más condiciones a la vez?**

En este caso debemos utilizar los siguientes operadores lógicos:

- condición 1 **&** condición 2 -> se cumplen ambas condiciones a la vez
- condición 1 **|** condición 2 -> se cumple una u otra de las condiciones
- condición 1 **&** ! condición 2 -> se cumple la condición 1 pero no la condición 2
- !condición 1 **&** condición 2 -> no se cumple la condición 1 pero si la condición 2
- !(condición 1 **&** condición 2) -> no se cumple ninguna de las 2 condiciones

Por ejemplo, si queremos filtrar todos los registros pertenecientes a la Comuna 5 y a la Comuna 13:

```
filtro <- filter(datos_amba, partido=="Comuna 5" & partido=="Comuna 13")

head(filtro)

## [1] created_on      provincia     partido      rooms
## [5] surface_total   surface_covered price        currency
## [9] title          property_type  operation_type lat
## [13] lon
## <0 rows> (or 0-length row.names)
```

El resultado es 0 porque un registro no puede pertenecer a ambas comunas al mismo tiempo, sin embargo si queremos filtrar aquellos que pertenecen a una u otra podemos hacerlo así:

```
filtro <- filter(datos_amba, partido=="Comuna 5" | partido=="Comuna 13")

head(filtro)

##   created_on provincia partido rooms surface_total surface_covered price
## 1 202006      CABA Comuna 13    1           30       30 18000
## 2 202006      CABA Comuna 13    1           31       29 17900
## 3 202006      CABA Comuna 13    2           60       50 32000
## 4 202006      CABA Comuna 13    2           53       44 26000
## 5 202006      CABA Comuna 5     2           45       38 20000
## 6 202006      CABA Comuna 13    2           55       55 45000
##   currency
## 1 ARS           Retasado! Monoambiente en Nuñez, excelente ubicación!
## 2 ARS           Departamento - Belgrano
## 3 ARS Depto 2AMB c/ Suite y Toilette - Deheza 1600 - Incluye Cochera
## 4 ARS           Unico 2 ambientes A Estrenar! Posesion inmediata! - Nuñez
## 5 ARS           Departamento de 2 ambientes en Alquiler en Almagro
## 6 ARS           BLANCO ENCALADA AL 3000 A ESTRENAR
##   title
## 1 Retasado! Monoambiente en Nuñez, excelente ubicación!
## 2 Departamento - Belgrano
## 3 Depto 2AMB c/ Suite y Toilette - Deheza 1600 - Incluye Cochera
## 4 Unico 2 ambientes A Estrenar! Posesion inmediata! - Nuñez
## 5 Departamento de 2 ambientes en Alquiler en Almagro
## 6 BLANCO ENCALADA AL 3000 A ESTRENAR
##   property_type operation_type      lat      lon
## 1 Departamento    Alquiler -34.55460 -58.46652
## 2 Departamento    Alquiler -34.56318 -58.46461
## 3 Departamento    Alquiler -34.53795 -58.46671
## 4 Departamento    Alquiler -34.54756 -58.47037
## 5 Departamento    Alquiler -34.60613 -58.42947
## 6 Departamento    Alquiler -34.56313 -58.46550
```

También podemos quedarnos con aquellas propiedades que estén en alquiler y que tengan más de 50m<sup>2</sup>:

```

filtro <- filter(datos_amba, operation_type=="Alquiler" & surface_covered>=50)

head(filtro)

##   created_on provincia partido rooms surface_total surface_covered price
## 1    202006      CABA Comuna 13     2           60             50 32000
## 2    202006      CABA Comuna 13     2           55             55 45000
## 3    202006      GBA  La Plata     2           50             50 18000
## 4    202006      GBA Ituzaingó    3           75             75 19000
## 5    202006      GBA    Tigre     3           95             80 42000
## 6    202006      GBA    Tigre     3           98             90 42000
##   currency
## 1     ARS
## 2     ARS
## 3     ARS
## 4     ARS
## 5     ARS
## 6     ARS
##
##                                     t
## 1                               Depto 2AMB c/ Suite y Toilette - Deheza 1600 - Incluye Cooc
## 2                                         BLANCO ENCALADA AL 3000 A ESTRE
## 3                               Alquiler, depto de un domitorio frente al parque San Ma
## 4                               Americana a 6 cuadras de la estacion de S.A. de P
## 5                               3 AMBIENTES EN EDIFICIO MIRADORES DE LA BAHIA CON COCHERA
## 6 ;EN EXCLUSIVA! Departamento en alquiler de 3 ambientes Miradores de La Bahía, Bahía de Norde
##   property_type operation_type      lat      lon
## 1 Departamento        Alquiler -34.53795 -58.46671
## 2 Departamento        Alquiler -34.56313 -58.46550
## 3 Departamento        Alquiler -34.93415 -57.96605
## 4       Casa            Alquiler -34.65957 -58.69712
## 5 Departamento        Alquiler -34.39642 -58.64665
## 6 Departamento        Alquiler -34.39512 -58.64665

```

O con aquellas propiedades que se ubiquen en la Comuna 12, 13 o 14 y que no sean monoambientes:

```

filtro <- filter(datos_amba, partido %in% c("Comuna 14", "Comuna 13", "Comuna 12") & ! rooms==1)

head(filtro)

##   created_on provincia partido rooms surface_total surface_covered price
## 1    202006      CABA Comuna 13     2           60             50 32000
## 2    202006      CABA Comuna 13     2           53             44 26000
## 3    202006      CABA Comuna 12     2           42             37 22000

```

```

## 4    202006    CABA Comuna 14    2        43      33 27000
## 5    202006    CABA Comuna 13    2        55      55 45000
## 6    202006    CABA Comuna 14    2        48      48 23000
## currency
## 1      ARS Depto 2AMB c/ Suite y Toilette - Deheza 1600 - Incluye Cochera
## 2      ARS         Unico 2 ambientes A Estrenar! Posesion inmediata! - Nuñez
## 3      ARS                     Depto 2AMB - Av. Balbin 3400 - ALQUILER
## 4      ARS             Dos ambientes con Balcón a Demaría y Patio.
## 5      ARS           BLANCO ENCALADA AL 3000 A ESTRENAR
## 6      ARS          Departamento - Belgrano
## property_type operation_type      lat      lon
## 1 Departamento     Alquiler -34.53795 -58.46671
## 2 Departamento     Alquiler -34.54756 -58.47037
## 3 Departamento     Alquiler -34.55700 -58.47898
## 4 Departamento     Alquiler -34.57421 -58.42243
## 5 Departamento     Alquiler -34.56313 -58.46550
## 6 Departamento     Alquiler -34.56456 -58.43559

```

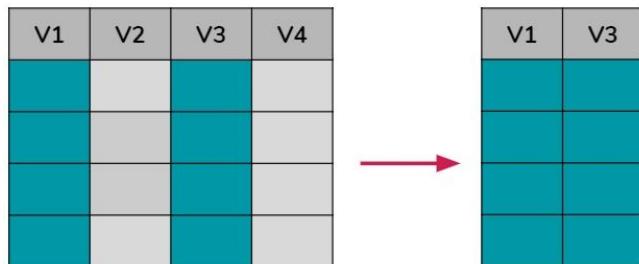
¡Y así se pueden hacer todas las combinaciones de filtros que queramos!

¿Y si en vez de quitar filas queremos quitar columnas? Bueno, aquí tenemos que usar la función de “seleccionar” que veremos a continuación.

## 2.2 Seleccionar columnas de interés

La función `select()` nos permite elegir u ordenar columnas de nuestro dataset. Esto se puede hacer indicando los nombres completos de las columnas, palabras que contienen, o la letra con la que empiezan o terminan.

Graficamente sería algo así:



Por ejemplo, de la siguiente forma podríamos quedarnos solo con las columnas `created_on`, `provincia`, `price` y `currency`:

```

seleccion <- select(datos_amba, created_on, provincia, price, currency)

head(seleccion)

```

```
##   created_on provincia price currency
## 1    202006      CABA 22500     ARS
## 2    202006      CABA 18000     ARS
## 3    202006      CABA 17900     ARS
## 4    202006      CABA 42000     ARS
## 5    202006      GBA 19000     ARS
## 6    202006      GBA 12000     ARS
```

También podríamos elegir que columna/s no queremos tener más en nuestro dataset agregando un “-” antes de su nombre:

```
seleccion <- select(datos_amba, -title)

head(seleccion)

##   created_on provincia      partido rooms surface_total surface_covered price
## 1    202006      CABA      Comuna 7     1           40              37 22500
## 2    202006      CABA      Comuna 13    1           30              30 18000
## 3    202006      CABA      Comuna 13    1           31              29 17900
## 4    202006      CABA      Comuna 1     1           35              35 42000
## 5    202006      GBA  Vicente López  1           36              27 19000
## 6    202006      GBA  La Matanza    2           24              24 12000
##   currency property_type operation_type      lat      lon
## 1     ARS  Departamento    Alquiler -34.61917 -58.46222
## 2     ARS  Departamento    Alquiler -34.55460 -58.46652
## 3     ARS  Departamento    Alquiler -34.56318 -58.46461
## 4     ARS  Departamento    Alquiler -34.61836 -58.36090
## 5     ARS  Departamento    Alquiler -34.53344 -58.49345
## 6     ARS          PH        Alquiler -34.66253 -58.52914
```

Con “:” podríamos indicar que queremos seleccionar un rango de columnas. Desde price hasta operation\_type:

```
seleccion <- select(datos_amba, price:operation_type)

head(seleccion)

##   price currency
## 1 22500     ARS
## 2 18000     ARS
## 3 17900     ARS
## 4 42000     ARS
## 5 19000     ARS
## 6 12000     ARS
```

```

##                                     title
## 1                               Departamento - Flores
## 2           Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3                               Departamento - Belgrano
## 4           Monoambiente con cochera. Zencity. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6                               PH - Lomas Del Mirador

##   property_type operation_type
## 1   Departamento      Alquiler
## 2   Departamento      Alquiler
## 3   Departamento      Alquiler
## 4   Departamento      Alquiler
## 5   Departamento      Alquiler
## 6           PH          Alquiler

```

O con las que ocupan de la posición 7 a la 10:

```
seleccion <- select(datos_amba, 7:10)
```

```
head(seleccion)
```

```

##   price currency
## 1 22500     ARS
## 2 18000     ARS
## 3 17900     ARS
## 4 42000     ARS
## 5 19000     ARS
## 6 12000     ARS

##                                     title
## 1                               Departamento - Flores
## 2           Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3                               Departamento - Belgrano
## 4           Monoambiente con cochera. Zencity. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6                               PH - Lomas Del Mirador

##   property_type
## 1   Departamento
## 2   Departamento
## 3   Departamento
## 4   Departamento
## 5   Departamento
## 6           PH

```

O agregando un “-” adelante podríamos quedarnos con aquellas que no ocupan de la posición 7 a 10:

```
seleccion <- select(datos_amba, -(7:10))

head(seleccion)

##   created_on provincia      partido rooms surface_total surface_covered
## 1    202006      CABA     Comuna 7     1          40            37
## 2    202006      CABA     Comuna 13    1          30            30
## 3    202006      CABA     Comuna 13    1          31            29
## 4    202006      CABA     Comuna 1     1          35            35
## 5    202006      GBA  Vicente López    1          36            27
## 6    202006      GBA  La Matanza     2          24            24
##   operation_type      lat      lon
## 1       Alquiler -34.61917 -58.46222
## 2       Alquiler -34.55460 -58.46652
## 3       Alquiler -34.56318 -58.46461
## 4       Alquiler -34.61836 -58.36090
## 5       Alquiler -34.53344 -58.49345
## 6       Alquiler -34.66253 -58.52914
```

Otra opción es seleccionar columnas de acuerdo a la primer letra de los nombres. Por ejemplo aquellas que comienzan con la letra “p”:

```
seleccion <- select(datos_amba, starts_with("p"))

head(seleccion)

##   provincia      partido price property_type
## 1      CABA     Comuna 7 22500  Departamento
## 2      CABA     Comuna 13 18000  Departamento
## 3      CABA     Comuna 13 17900  Departamento
## 4      CABA     Comuna 1 42000  Departamento
## 5      GBA  Vicente López 19000  Departamento
## 6      GBA  La Matanza 12000           PH
```

O aquellas que sus nombres terminan con la letra “e”:

```
seleccion <- select(datos_amba, ends_with("e"))

head(seleccion)

##   price
## 1 22500
## 2 18000
```

```

## 3 17900
## 4 42000
## 5 19000
## 6 12000
##
## title
## 1 Departamento - Flores
## 2 Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3 Departamento - Belgrano
## 4 Monoambiente con cochera. Zencyty. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6 PH - Lomas Del Mirador
##   property_type operation_type
## 1 Departamento      Alquiler
## 2 Departamento      Alquiler
## 3 Departamento      Alquiler
## 4 Departamento      Alquiler
## 5 Departamento      Alquiler
## 6          PH        Alquiler

```

O que sus nombres contengan la palabra “surface”:

```

seleccion <- select(datos_amba, contains("surface"))

head(seleccion)

```

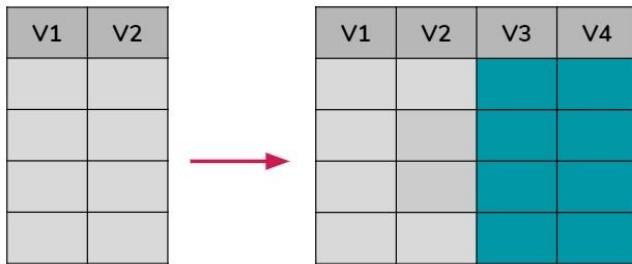
```

##   surface_total surface_covered
## 1           40            37
## 2           30            30
## 3           31            29
## 4           35            35
## 5           36            27
## 6           24            24

```

## 2.3 Modificar o agregar columnas

Ahora veamos como mutar nuestro dataset agregando nuevas columnas o cambiando el contenido de las existentes. Gráficamente sería algo así:



Aprovechando que tenemos los datos del precio total (price) y superficie cubierta (surface\_covered) de cada una de las propiedades, agreguemos una nueva columna a nuestro dataset que incluya el valor del m2:

```
modificar <- mutate(datos_amba, price_m2=price/surface_covered)

head(modificar)
```

```
##   created_on provincia      partido rooms surface_total surface_covered price
## 1 202006      CABA Comuna 7     1        40            37  22500
## 2 202006      CABA Comuna 13    1        30            30  18000
## 3 202006      CABA Comuna 13    1        31            29  17900
## 4 202006      CABA Comuna 1     1        35            35  42000
## 5 202006      GBA Vicente López 1        36            27  19000
## 6 202006      GBA La Matanza   2        24            24  12000
##   currency
## 1 ARS
## 2 ARS
## 3 ARS
## 4 ARS
## 5 ARS
## 6 ARS
##                                title
## 1                     Departamento - Flores
## 2 Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3                     Departamento - Belgrano
## 4 Monoambiente con cochera. Zencyt. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6                      PH - Lomas Del Mirador
##   property_type operation_type      lat      lon price_m2
## 1 Departamento     Alquiler -34.61917 -58.46222  608.1081
## 2 Departamento     Alquiler -34.55460 -58.46652  600.0000
## 3 Departamento     Alquiler -34.56318 -58.46461  617.2414
## 4 Departamento     Alquiler -34.61836 -58.36090 1200.0000
## 5 Departamento     Alquiler -34.53344 -58.49345  703.7037
## 6             PH     Alquiler -34.66253 -58.52914  500.0000
```

Si queremos redondear el resultado obtenido, principalmente cuando es una división y por default nos pone varios decimales, tenemos que usar `round()` y asignar la cantidad de decimales deseados, en este caso usaré 2:

```
modificar <- mutate(datos_amba, price_m2=round(price/surface_covered, 2))

head(modificar)

##   created_on provincia      partido rooms surface_total surface_covered price
## 1 202006      CABA     Comuna 7     1          40            37 22500
## 2 202006      CABA     Comuna 13    1          30            30 18000
## 3 202006      CABA     Comuna 13    1          31            29 17900
## 4 202006      CABA     Comuna 1     1          35            35 42000
## 5 202006      GBA      Vicente López 1          36            27 19000
## 6 202006      GBA      La Matanza   2          24            24 12000
##   currency
## 1 ARS
## 2 ARS
## 3 ARS
## 4 ARS
## 5 ARS
## 6 ARS
##
##   title
## 1 Departamento - Flores
## 2 Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3 Departamento - Belgrano
## 4 Monoambiente con cochera. Zencyt. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6 PH - Lomas Del Mirador
##   property_type operation_type      lat      lon price_m2
## 1 Departamento     Alquiler -34.61917 -58.46222  608.11
## 2 Departamento     Alquiler -34.55460 -58.46652  600.00
## 3 Departamento     Alquiler -34.56318 -58.46461  617.24
## 4 Departamento     Alquiler -34.61836 -58.36090 1200.00
## 5 Departamento     Alquiler -34.53344 -58.49345  703.70
## 6             PH     Alquiler -34.66253 -58.52914  500.00
```

Como verán, para hacer cálculos entre columnas numéricas podemos utilizar:

- A / B **A dividido B**
- A \* B **A multiplicado por B**
- A + B **Suma de A y B**
- A - B **Resta de A menos B**

También podríamos agregar una columna que refleje un cálculo entre una columna existente y un valor extra, por ejemplo pasemos la superficie total de m<sup>2</sup> a cm<sup>2</sup>:

```
modificar <- mutate(datos_amba, surface_total_cm2=surface_total*10000)

head(modificar)

##   created_on provincia      partido rooms surface_total surface_covered price
## 1 202006       CABA     Comuna 7     1           40             37 22500
## 2 202006       CABA     Comuna 13    1           30             30 18000
## 3 202006       CABA     Comuna 13    1           31             29 17900
## 4 202006       CABA     Comuna 1     1           35             35 42000
## 5 202006       GBA  Vicente López    1           36             27 19000
## 6 202006       GBA  La Matanza     2           24             24 12000

##   currency
## 1      ARS
## 2      ARS
## 3      ARS
## 4      ARS
## 5      ARS
## 6      ARS

##   title
## 1          Departamento - Flores
## 2 Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3          Departamento - Belgrano
## 4          Monoambiente con cochera. Zencytity. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6          PH - Lomas Del Mirador

##   property_type operation_type      lat      lon surface_total_cm2
## 1 Departamento      Alquiler -34.61917 -58.46222        400000
## 2 Departamento      Alquiler -34.55460 -58.46652        300000
## 3 Departamento      Alquiler -34.56318 -58.46461        310000
## 4 Departamento      Alquiler -34.61836 -58.36090        350000
## 5 Departamento      Alquiler -34.53344 -58.49345        360000
## 6          PH      Alquiler -34.66253 -58.52914        240000
```

Dejando de lado los cálculos, otra posibilidad que tenemos es separar el contenido de una columna en 2, por ejemplo dividamos en año y mes la data que aparece en created\_on. Para esto utilizaremos substr():

```
modificar <- mutate(datos_amba,
                     year = substr(created_on, 1, 4),
                     month = substr(created_on, 5, 6))

head(modificar)
```

```

##   created_on provincia      partido rooms surface_total surface_covered price
## 1    202006     CABA     Comuna 7     1          40            37 22500
## 2    202006     CABA     Comuna 13    1          30            30 18000
## 3    202006     CABA     Comuna 13    1          31            29 17900
## 4    202006     CABA     Comuna 1     1          35            35 42000
## 5    202006     GBA  Vicente López    1          36            27 19000
## 6    202006     GBA  La Matanza     2          24            24 12000
##   currency
## 1     ARS
## 2     ARS
## 3     ARS
## 4     ARS
## 5     ARS
## 6     ARS
##
##                                         title
## 1                         Departamento - Flores
## 2             Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3                         Departamento - Belgrano
## 4           Monoambiente con cochera. Zencity. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6                           PH - Lomas Del Mirador
##   property_type operation_type      lat      lon year month
## 1 Departamento       Alquiler -34.61917 -58.46222 2020     06
## 2 Departamento       Alquiler -34.55460 -58.46652 2020     06
## 3 Departamento       Alquiler -34.56318 -58.46461 2020     06
## 4 Departamento       Alquiler -34.61836 -58.36090 2020     06
## 5 Departamento       Alquiler -34.53344 -58.49345 2020     06
## 6           PH         Alquiler -34.66253 -58.52914 2020     06

```

Otra aplicación que tiene `mutate()` es la de agregar columnas con algún contenido que elijamos nosotros, como por ejemplo sumemos una nueva columna que indique la fuente de donde descargamos toda esta información:

```
modificar <- mutate(datos_amba, fuente="Properati")
```

```
head(modificar)
```

```

##   created_on provincia      partido rooms surface_total surface_covered price
## 1    202006     CABA     Comuna 7     1          40            37 22500
## 2    202006     CABA     Comuna 13    1          30            30 18000
## 3    202006     CABA     Comuna 13    1          31            29 17900
## 4    202006     CABA     Comuna 1     1          35            35 42000
## 5    202006     GBA  Vicente López    1          36            27 19000
## 6    202006     GBA  La Matanza     2          24            24 12000
##   currency

```

```

## 1      ARS
## 2      ARS
## 3      ARS
## 4      ARS
## 5      ARS
## 6      ARS
##
##                                     title
## 1                               Departamento - Flores
## 2           Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3                               Departamento - Belgrano
## 4           Monoambiente con cochera. Zencyty. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6                               PH - Lomas Del Mirador
##   property_type operation_type      lat      lon     fuente
## 1 Departamento      Alquiler -34.61917 -58.46222 Properati
## 2 Departamento      Alquiler -34.55460 -58.46652 Properati
## 3 Departamento      Alquiler -34.56318 -58.46461 Properati
## 4 Departamento      Alquiler -34.61836 -58.36090 Properati
## 5 Departamento      Alquiler -34.53344 -58.49345 Properati
## 6          PH        Alquiler -34.66253 -58.52914 Properati

```

Ahora veamos como modificar el tipo de dato dentro de una columna:

```

class(datos_amba$title)

## [1] "factor"

```

Vemos que la variable “title” es de tipo factor, así que cambiemos su formato a character:

```

modificar <- mutate(datos_amba, title=as.character(title))

class(modificar$title)

## [1] "character"

```

En el ejemplo anterior utilizamos `as.character()` pero si quisiesemos convertir una variable a factor utilizariamos `as.factor()`, a numérica `as.numeric()` o a número entero `as.integer()`,

Por último, veamos como unir 2 columnas de texto en una con `paste()`:

```

modificar <- mutate(datos_amba, prov_partido=paste(provincia, partido, sep="_"))

head(modificar)

##   created_on provincia      partido rooms surface_total surface_covered price
## 1 202006      CABA     Comuna 7     1          40            37 22500
## 2 202006      CABA     Comuna 13    1          30            30 18000
## 3 202006      CABA     Comuna 13    1          31            29 17900
## 4 202006      CABA     Comuna 1     1          35            35 42000
## 5 202006      GBA  Vicente López    1          36            27 19000
## 6 202006      GBA  La Matanza     2          24            24 12000
##   currency
## 1      ARS
## 2      ARS
## 3      ARS
## 4      ARS
## 5      ARS
## 6      ARS
##
##                                     title
## 1                         Departamento - Flores
## 2             Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3                         Departamento - Belgrano
## 4           Monoambiente con cochera. Zencity. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6                           PH - Lomas Del Mirador
##   property_type operation_type      lat      lon prov_partido
## 1 Departamento       Alquiler -34.61917 -58.46222  CABA_Comuna 7
## 2 Departamento       Alquiler -34.55460 -58.46652  CABA_Comuna 13
## 3 Departamento       Alquiler -34.56318 -58.46461  CABA_Comuna 13
## 4 Departamento       Alquiler -34.61836 -58.36090  CABA_Comuna 1
## 5 Departamento       Alquiler -34.53344 -58.49345 GBA_Vicente López
## 6             PH       Alquiler -34.66253 -58.52914  GBA_La Matanza

```

## 2.4 Ordenar registros

Esta función nos permitirá ordenar las columnas en orden ascendente o descendente como se ve a continuación:



Probemos ordenar las filas de nuestro data frame en función de los valores de una o más columnas. Por defecto se ordena en forma ascendente:

```
ordenar <- arrange(datos_amba, surface_total)

head(ordenar)
```

```
##   created_on provincia partido rooms surface_total surface_covered price
## 1 202007      CABA Comuna 15     1          10           10 25000
## 2 202006      GBA  Tigre        2          12           12 10500
## 3 202007      CABA Comuna 1      1          14           14 35000
## 4 202006      CABA Comuna 2      1          15           15 12900
## 5 202007      CABA Comuna 15     1          18           18 48000
## 6 202006      GBA  Morón        1          18           18 35000
##   currency
## 1 USD
## 2 ARS
## 3 USD
## 4 ARS
## 5 USD
## 6 USD
##
##                                     title
## 1 Garage - Cochera Descubierta en Villa Crespo
## 2 Departamento de 2 ambientes en Pradera, Santa Barbara
## 3 VENTA PH 13,84 M2 PANTA BAJA BALVANERA
## 4 Departamento en Alquiler en Barrio Norte
## 5 Venta Monoambiente estudio con amenities Chacarita
## 6 Lindo monoambiente en 2do piso por escalera al frente. BAJAS EXPENSAS!!!
##   property_type operation_type      lat      lon
## 1 Departamento          Venta -34.60330 -58.45547
## 2 Departamento          Alquiler -34.44604 -58.63236
## 3 PH                  Venta -34.61566 -58.39221
## 4 Departamento          Alquiler -34.59092 -58.40632
## 5 Departamento          Venta -34.59043 -58.44666
## 6 Departamento          Venta -34.64390 -58.63221
```

Pero si queremos ordenar en forma descendente debemos aclararlo con `desc()`:

```

ordenar <- arrange(datos_amba, desc(surface_total))

head(ordenar)

##   created_on provincia     partido rooms surface_total surface_covered price
## 1    202007      GBA       Pilar     5        5000            300 120000
## 2    202006      GBA San Vicente    3        4356            150  85000
## 3    202007      GBA       Pilar     6        3780            320 220000
## 4    202006      CABA  Comuna 13    2        3650            35 109500
## 5    202007      GBA      Luján     4        2703            182 370000
## 6    202007      GBA       Pilar     3        2500             70 121000
##   currency
## 1      USD
## 2      USD España entre Buchard y Esmeralda. Casa en venta, San Vicente
## 3      USD Venta casaquinta Del Viso Residencial Los Jazmines
## 4      USD Acogedor Departamento de dos ambientes en Belgrano
## 5      USD Casa 3 dormitorios estilo campo Estancias Golf
## 6      USD VENTA CASA 2 AMBIENTES PILAR 2500 M2 DE PARQUE!!
##   property_type operation_type      lat      lon
## 1          Casa      Venta -34.40066 -58.86919
## 2          Casa      Venta -35.02984 -58.40594
## 3          Casa      Venta -34.45687 -58.79651
## 4  Departamento      Venta -34.55398 -58.44853
## 5          Casa      Venta -34.49543 -59.00930
## 6          Casa      Venta -34.41586 -58.85729

```

También podemos ordenar por 2 o más columnas. En este caso, R priorizará ordenar la primera, luego la segunda, y así sucesivamente. Veamos un ejemplo:

```

ordenar <- arrange(datos_amba, partido, rooms)

head(ordenar)

```

```

##   created_on provincia     partido rooms surface_total surface_covered
## 1    202006      GBA Almirante Brown    2           40            40
## 2    202006      GBA Almirante Brown    2           60            60
## 3    202006      GBA Almirante Brown    2           50            50
## 4    202007      GBA Almirante Brown    2           52            48
## 5    202007      GBA Almirante Brown    2           47            47
## 6    202007      GBA Almirante Brown    2           50            50
##   price currency
## 1 14500      ARS
## 2 20000      ARS
## 3 125000     USD

```

```

## 4 18000     ARS
## 5 16000     ARS
## 6 96000     USD
##
##                                     title
## 1                               Departamento en alquiler
## 2                               Departamento - Adrogué
## 3                               Departamento - Adrogue
## 4 Venta de Departamento 2 ambientes en Hermoso Edificio a metros de Plaza Brown de Adrogué
## 5                               DEPARTAMENTO 2 AMBIENTES EN AV ESPORA AL 800
## 6                               Departamento - Adrogué
##   property_type operation_type      lat      lon
## 1 Departamento       Alquiler -34.81144 -58.39264
## 2 Departamento       Alquiler -34.79964 -58.38278
## 3 Departamento       Venta  -34.79732 -58.39132
## 4 Departamento       Alquiler -34.79720 -58.38280
## 5 Departamento       Alquiler -34.79880 -58.38863
## 6 Departamento       Venta  -34.79913 -58.38364

ordenar <- arrange(datos_amba, partido, desc(rooms))

head(ordenar)

##   created_on provincia      partido rooms surface_total surface_covered
## 1    202007      GBA Almirante Brown     8        447          189
## 2    202006      GBA Almirante Brown     7        670          590
## 3    202006      GBA Almirante Brown     5        239          112
## 4    202006      GBA Almirante Brown     5        260          120
## 5    202007      GBA Almirante Brown     5        370          220
## 6    202006      GBA Almirante Brown     5        180          180
##   price currency
## 1 230000     USD
## 2 1500000    USD
## 3 169000     USD
## 4 248000     USD
## 5 230000     USD
## 6 390000     USD
##
##                                     title
## 1                           Casa c/2 dptos, local y galpón - Lomas del Mirador
## 2                           Casa - Adrogué
## 3 A REFACCIONAR! Casona colonial con gran lote, fondo libre, parrilla, 3 dormitorios 2 baños
## 4                               Hermoso Duplex 3 dormitorios gran patio 4 cocheras
## 5                               Chalet en venta Barrio Santa Rita - Longchamps
## 6 Casa en Construcción en Venta en Barrio Privado 'Brisas de Adrogué'
##   property_type operation_type      lat      lon
## 1       Casa         Venta -34.86568 -58.37823

```

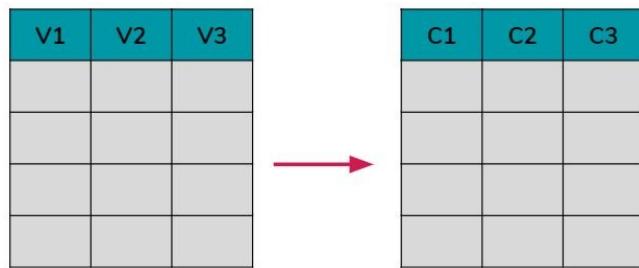
```

## 2      Casa      Venta -34.80905 -58.40631
## 3      PH       Venta -34.79638 -58.38691
## 4      Casa      Venta -34.79746 -58.37885
## 5      Casa      Venta -34.86473 -58.39129
## 6      Casa      Venta -34.81368 -58.40300

```

## 2.5 Renombrar columnas

Ahora veamos como cambiar los nombres a una o más columnas existentes en nuestro dataset:



Empecemos cambiando el nombre de la variable “rooms” por “ambientes”:

```

renombrar <- rename(datos_amba, ambientes=rooms)

head(renombrar)

```

```

##   created_on provincia      partido ambientes surface_total surface_covered
## 1 202006      CABA     Comuna 7      1          40            37
## 2 202006      CABA     Comuna 13     1          30            30
## 3 202006      CABA     Comuna 13     1          31            29
## 4 202006      CABA     Comuna 1      1          35            35
## 5 202006      GBA      Vicente López 1          36            27
## 6 202006      GBA      La Matanza    2          24            24
##   price currency
## 1 22500      ARS
## 2 18000      ARS
## 3 17900      ARS
## 4 42000      ARS
## 5 19000      ARS
## 6 12000      ARS
## 
## 1
## 2
## 3
##   title
##   Departamento - Flores
##   Retasado! Monoambiente en Nuñez, excelente ubicación!
##   Departamento - Belgrano

```

```

## 4           Monoambiente con cochera. Zencity. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6                                     PH - Lomas Del Mirador
##   property_type operation_type      lat      lon
## 1 Departamento     Alquiler -34.61917 -58.46222
## 2 Departamento     Alquiler -34.55460 -58.46652
## 3 Departamento     Alquiler -34.56318 -58.46461
## 4 Departamento     Alquiler -34.61836 -58.36090
## 5 Departamento     Alquiler -34.53344 -58.49345
## 6          PH     Alquiler -34.66253 -58.52914

```

Como habrán notado, primero hay que poner el nombre de la nueva columna y luego el de la columna actual. Esto es muy importante, porque si lo hacemos al revés nos dará un error.

Ahora veamos un ejemplo y cambiemos los nombres de 3 columnas:

```

renombrar <- rename(datos_amba, ambientes=rooms, m2_cubierto=surface_covered, m2_total=surface_to
head(renombrar)

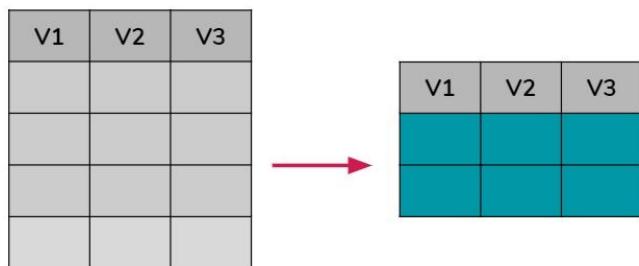
##   created_on provincia      partido ambientes m2_total m2_cubierto price
## 1 202006      CABA      Comuna 7       1      40        37 22500
## 2 202006      CABA      Comuna 13      1      30        30 18000
## 3 202006      CABA      Comuna 13      1      31        29 17900
## 4 202006      CABA      Comuna 1       1      35        35 42000
## 5 202006      GBA       Vicente López    1      36        27 19000
## 6 202006      GBA       La Matanza     2      24        24 12000
##   currency
## 1      ARS
## 2      ARS
## 3      ARS
## 4      ARS
## 5      ARS
## 6      ARS
##   title
## 1           Departamento - Flores
## 2 Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3           Departamento - Belgrano
## 4           Monoambiente con cochera. Zencity. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6                                     PH - Lomas Del Mirador
##   property_type operation_type      lat      lon
## 1 Departamento     Alquiler -34.61917 -58.46222
## 2 Departamento     Alquiler -34.55460 -58.46652
## 3 Departamento     Alquiler -34.56318 -58.46461

```

```
## 4 Departamento      Alquiler -34.61836 -58.36090
## 5 Departamento      Alquiler -34.53344 -58.49345
## 6                 PH    Alquiler -34.66253 -58.52914
```

## 2.6 Resumir y agrupar datos

Esta función es súper útil cuando manipulamos datos ya que nos permitirá realizar resumenes/sumarios de la data completa, obteniendo por ejemplo valores promedio, máximos o mínimos de una o más columnas.



Probemos calculando la mediana de todos los valores que aparecen en la columna surface\_covered:

```
summarise(datos_amba, surface_covered=median(surface_covered))
```

```
##   surface_covered
## 1             66
```

O calculemos un promedio de toda la columna surface\_total:

```
summarise(datos_amba, surface_covered=mean(surface_covered))
```

```
##   surface_covered
## 1            91.98151
```

También podemos averiguar el valor máximo o mínimo de alguna variable:

```
summarise(datos_amba, surface_covered=max(rooms))
```

```
##   surface_covered
## 1             10
```

```
summarise(datos_amba, surface_covered=min(rooms))

##   surface_covered
## 1               1
```

Como verán, esta función resulta útil para ver valores agregados de toda la base, sin embargo, también podemos agrupar los datos previo a calcular los resúmenes, y así obtener resúmenes por agrupaciones en vez de uno solo para toda la base. Para esto vamos a utilizar `summarise()` junto a `group_by()`. Veamos un ejemplo:

- Primero agrupemos los datos por la variable “`operation_type`”.
- Luego calculemos el promedio de superficie cubierta sobre la agrupación realizada previamente.

```
agrupar <- group_by(datos_amba, operation_type)

resumir <- summarise(agrupar, surface_covered=mean(surface_covered))

head(resumir)

## # A tibble: 2 x 2
##   operation_type surface_covered
##   <fct>           <dbl>
## 1 Alquiler        72.9
## 2 Venta           97.3
```

Con la agrupación y el resumen podemos ver que la superficie cubierta promedio de las propiedades en alquiler es 72,93m<sup>2</sup> y la de las propiedades en venta es 97,28m<sup>2</sup>.

Probemos agrupando por 3 columnas:

```
agrupar <- group_by(datos_amba, operation_type, currency)

resumir <- summarise(agrupar, price_m2=mean(price/surface_covered))

head(resumir)

## # A tibble: 2 x 3
## # Groups:   operation_type [2]
##   operation_type currency price_m2
##   <fct>          <fct>     <dbl>
## 1 Alquiler        ARS       520.
## 2 Venta           USD      2689.
```

El valor del m<sup>2</sup> promedio para las propiedades en Alquiler es de 520 \$ARS mientras que para las propiedades en venta es de 2.689 USD. Sin embargo, está claro que en el valor de CABA y GBA es diferente, así que calculemos el valor del m<sup>2</sup> promedio para cada uno:

```
agrupar <- group_by(datos_amba, operation_type, currency, provincia)
resumir <- summarise(agrupar, rooms=mean(price/surface_covered))

head(resumir)

## # A tibble: 4 x 4
## # Groups:   operation_type, currency [2]
##   operation_type currency provincia rooms
##   <fct>        <fct>    <fct>     <dbl>
## 1 Alquiler      ARS      CABA      617.
## 2 Alquiler      ARS      GBA       399.
## 3 Venta         USD      CABA     3119.
## 4 Venta         USD      GBA      2024.
```

Se ve claramente que hay una diferencia entre CABA y GBA, siendo CABA más caro para ambos tipos de operación. Por último probemos desagregando esta información por partido:

```
agrupar <- group_by(datos_amba, operation_type, currency, partido)
resumir <- summarise(agrupar, price_m2=mean(price/surface_covered))

head(resumir)

## # A tibble: 6 x 4
## # Groups:   operation_type, currency [1]
##   operation_type currency partido      price_m2
##   <fct>        <fct>    <fct>     <dbl>
## 1 Alquiler      ARS      Almirante Brown 304.
## 2 Alquiler      ARS      Avellaneda     299.
## 3 Alquiler      ARS      Berazategui   311.
## 4 Alquiler      ARS      Comuna 1      699.
## 5 Alquiler      ARS      Comuna 10     381.
## 6 Alquiler      ARS      Comuna 11     444.
```

## 2.7 Concatenar funciones (%>%)

Llegamos al final de la clase, ya vimos varias funciones por separado, pero ¿Qué pasa si queremos aplicarlas todas a la vez? ¿Cómo podemos hacerlo?

En este caso debemos usar el operador pipe (%>%) **Ctrl+Shift+M** que sirve para encadenar funciones, y en vez de realizar una por una, poder realizar todas juntas.

Veamos algunos ejemplos:

Imaginemos que queremos calcular por mes y para cada provincia (CABA y GBA) cuantas propiedades hubo en venta, con que valor total promedio, con que valor del m<sup>2</sup> promedio y con que superficie promedio. Así lo tenemos que hacer según lo aprendido hasta ahora:

```
concatenar <- filter(datos_amba, operation_type=="Venta")

concatenar <- select(concatenar, created_on, provincia, surface_covered, price)

concatenar <- group_by(concatenar, created_on, provincia)

concatenar <- summarise(concatenar, cantidad=n(),
                        price=mean(price),
                        surface_covered=mean(surface_covered),
                        price_m2=price/surface_covered)

head(concatenar)

## # A tibble: 4 x 6
## # Groups:   created_on [2]
##   created_on provincia cantidad  price surface_covered price_m2
##       <int>     <fct>    <int>    <dbl>        <dbl>    <dbl>
## 1     202006    CABA      3986  291547.       85.3     3420.
## 2     202006    GBA       2772  231074.      121.     1903.
## 3     202007    CABA      3110  251563.       80.5     3124.
## 4     202007    GBA       1810  206180.      116.     1783.
```

Y así lo deberíamos hacer con pipe %>%:

```
concatenar <- datos_amba %>%
  filter(operation_type=="Venta") %>%
  select(created_on, provincia, surface_covered, price) %>%
  group_by(created_on, provincia) %>%
  summarise(cantidad=n(),
            price=mean(price),
            surface_covered=mean(surface_covered),
            price_m2=price/surface_covered)

head(concatenar)
```

```
## # A tibble: 4 x 6
## # Groups:   created_on [2]
##   created_on provincia cantidad  price surface_covered price_m2
##   <int>     <fct>    <int>   <dbl>        <dbl>      <dbl>
## 1 202006  CABA       3986 291547.     85.3     3420.
## 2 202006  GBA        2772 231074.    121.     1903.
## 3 202007  CABA       3110 251563.     80.5     3124.
## 4 202007  GBA        1810 206180.    116.     1783.
```

A partir de la agrupación y resumen por ejemplo podemos ver que:

- En ambos meses, en CABA hubo mayor cantidad de propiedades en venta.
- En ambos meses, la superficie cubierta de las propiedades en venta en GBA son mayores que las de CABA.
- En ambos meses el valor promedio del m<sup>2</sup> es más alto en CABA que en GBA.
- En ambas zonas (CABA y GBA), entre Junio y Julio 2020 hubo una caída en la cantidad de propiedades publicadas y en el valor promedio del m<sup>2</sup>.

Como verán, en ambos casos llegamos al mismo resultado, pero sin dudas, la segunda opción es la recomendable porque nos ahorraremos varias líneas de código y resultados intermedios.

## 2.8 Transformar la estructura de los datos

El paquete `tidyverse` también nos permite realizar transformaciones en la estructura de nuestro dataset. Pero, ¿A qué nos referimos con esto? A continuación analizaremos 2 casos:

Tipo 1. Nuestro dataset tiene **diferentes categorías** que están **separadas en muchas filas**. Por ejemplo:

COMUNA	INDICADOR	CANTIDAD
Comuna 13	POBLACION	230.763
Comuna 13	VIVIENDAS	129.564
Comuna 13	HOGARES	100.334
Comuna 7	POBLACION	220.591
Comuna 7	VIVIENDAS	89.688
Comuna 7	HOGARES	81.483

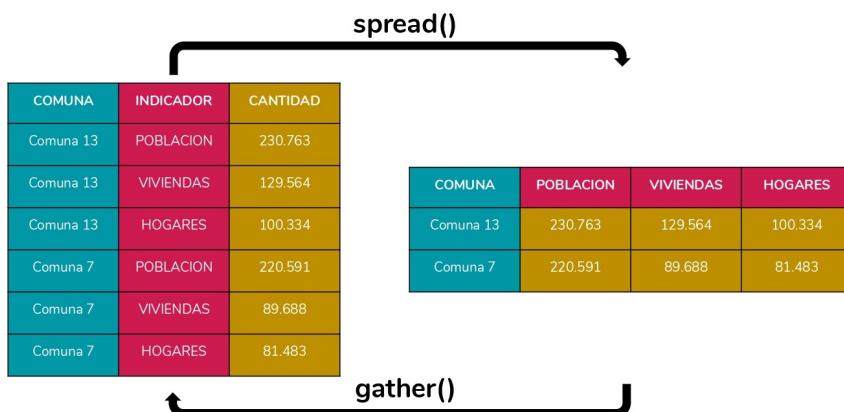
Tipo 2. Nuestro dataset tiene **diferentes categorías** que están **separadas en muchas columnas**. Por ejemplo:

COMUNA	POBLACION	VIVIENDAS	HOGARES
Comuna 13	230.763	129.564	100.334
Comuna 7	220.591	89.688	81.483

COMUNA	POBLACION	VIVIENDAS	HOGARES
Comuna 13	230.763	129.564	100.334
Comuna 7	220.591	89.688	81.483

Como verán, en ambos casos la información es la misma, lo que cambia es la estructura.

Es muy probable que al analizar bases de datos nos encontramos con algunas estructuras similares a estas y querremos cambiarlas. Para pasar del Tipo 1 de estructura al Tipo 2 y viceversa utilizaremos las funciones `spread()` y `gather()` respectivamente.



Sigamos utilizando nuestro dataset de Properati, pero para trabajar con menos volumen de datos y que sea más sencillo comprender el ejemplo, hagamos una agrupación por provincia y tipo de operación y calculemos el valor del m2:

```
transformar <- datos_amba %>%
  group_by(provincia, operation_type) %>%
  summarise(valor_m2=mean(price/surface_covered))
```

```
head(transformar)
```

```
## # A tibble: 4 x 3
## # Groups:   provincia [2]
##   provincia operation_type valor_m2
##   <fct>      <fct>        <dbl>
## 1 CABA       Alquiler     617.
## 2 CABA       Venta        3119.
## 3 GBA        Alquiler     399.
## 4 GBA        Venta        2024.
```

Bien, ya tenemos un pequeño dataset de 3 columnas y 4 filas que presentan una estructura similar a la que se muestra al inicio como “Tipo 1”.

### 2.8.1 Extender la estructura de los datos: spread()

Si necesitamos pasar el dataset al otro formato extendido, debemos usar `spread()` indicando el dataset que quiero modificar (`datos_amba`), la columna que quiero separar (`key=operation_type`) y la columna que contiene los valores que quiero mantener en la tabla (`value=valor_m2`):

```
extender <- spread(transformar, key=operation_type, value=valor_m2)
```

El resultado es el siguiente:

```
head(extender)
```

```
## # A tibble: 2 x 3
## # Groups:   provincia [2]
##   provincia Alquiler Venta
##   <fct>      <dbl> <dbl>
## 1 CABA       617.  3119.
## 2 GBA        399.  2024.
```

Se ve como ahora tenemos un dataset extendido, donde las categorías incluidas en columna `operation_type` se dividieron en 2 columnas: “Alquiler” y “Venta”. Pero, ¿Cómo hacemos si queremos volver al dataset original?

Esto lo hacemos con la función `gather()`.

### 2.8.2 Unificar los datos: gather()

Esta función que une múltiples columnas en una sola, se suele utilizar cuando todas las columnas representan valores de una variable. Por ejemplo, en nuestro caso, tanto “Alquiler” como “Venta” pueden ser categorías de una variable llamada `operation_type`.

Para utilizar `gather()` debemos indicar el dataset que queremos modificar (`datos_amba`), la nueva columna donde queremos agrupar múltiples columnas (`key="operation_type"`), la nueva columna donde queremos agrupar los valores de toda la tabla (`value="valor_m2"`), y por último debemos indicar que rango de columnas son las que queremos incluir en la key (`Alquiler:Venta`).

```
unificar <- gather(extender, key="operation_type", value="valor_m2", Alquiler:Venta)

head(unificar)

## # A tibble: 4 x 3
## # Groups:   provincia [2]
##   provincia operation_type valor_m2
##   <fct>     <chr>          <dbl>
## 1 CABA      Alquiler       617.
## 2 GBA       Alquiler       399.
## 3 CABA      Venta         3119.
## 4 GBA       Venta         2024.
```

Y ahora si, volvimos al otro formato de tabla.



## Capítulo 3

# ANÁLISIS Y VISUALIZACIÓN DE DATOS

El paquete `tidyverse` incluye diversos paquetes, entre los que se encuentra `ggplot2`, que nos permite realizar diferentes tipos de gráficos a partir de nuestros datos.

Hasta acá ya sabemos abrir, conocer, manipular y transformar un dataset, así que ahora nos enfocaremos en el **desarrollo de visualizaciones** que nos permitirán comunicar de forma gráfica lo que dicen nuestros datos.

En la estructura del código necesario para realizar un gráfico con `ggplot2` hay que determinar lo siguiente:

- **Dataset a utilizar:** hay que indicar el dataset al inicio del código en `ggplot(dataset)`.
- **Tipo de Gráfico** a realizar: puede ser de puntos, barras, líneas, áreas, matriz, histograma, densidad, etc. Estos elementos funcionan como “capas” ya que pueden utilizarse más de uno a la vez en un mismo gráfico y sus códigos se escriben `geom_hist()`, `geom_tile()`, `geom_bar()`, etc.
- **Atributos Estéticos** que dependen de una o más columnas del dataset: variables a utilizar en ejes x e y, colores, tamaños, formas, transparencias, etc. Siempre se asignan dentro de `aes()`.
- **Etiquetas** que ayudan a la interpretación de las visualizaciones. Estas son: títulos, subtítulos, leyendas, etc. Se asignan dentro de `labs()`.
- **Facetas** a utilizar: de acuerdo a alguna columna del dataset dividen el gráfico manteniendo las escalas. Esto es opcional y se hace con `facet_grid()`.

- **Temas** que permiten elegir la estética general del gráfico (color de fondo, tamaño de márgenes, etc). Algunos son `theme_void()`, `theme_dark()`, etc.

A continuación seguiremos trabajando con los datos de Properati de Junio y Julio 2020 en AMBA y desarrollaremos diferentes visualizaciones que sinteticen y comuniquen la información que contiene.

En particular, veremos como representar gráficamente lo siguiente:

- Distribución de una variable continua
- Distribución de valores continuos asociados a una variable categórica
- Relación entre variables continuas
- Relación entre variables categóricas
- Relación entre una variable continua y una categórica

Empecemos activando la librería `tidyverse`:

```
library(tidyverse)
```

Y volvamos a cargar nuestro dataset:

```
datos_amba <- read.csv("data/amba_properati.csv")
```

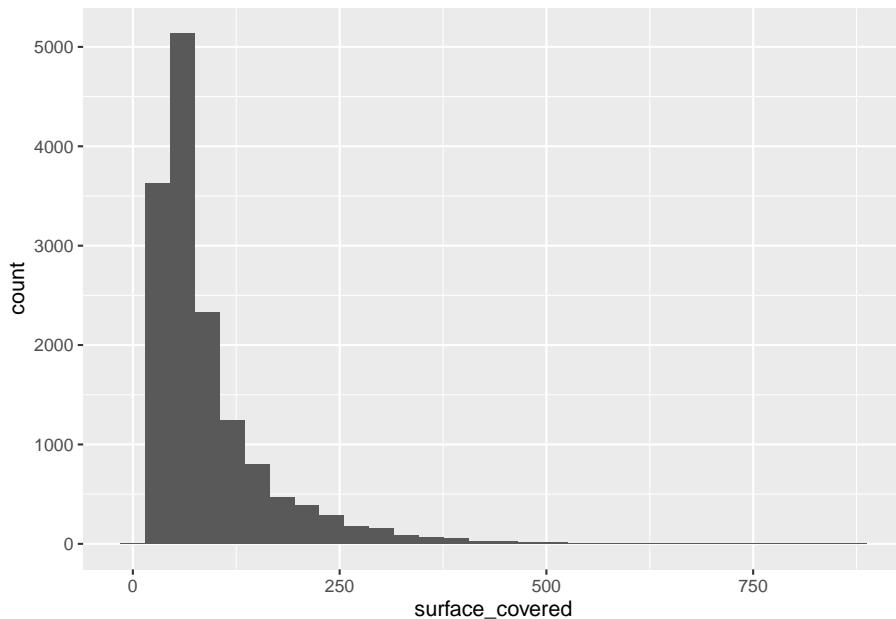
## 3.1 Distribución de una variable continua

### 3.1.1 Histograma

Los histogramas muestran gráficamente, a partir de barras, la distribución de una variable continua, es decir la frecuencia con la que aparece cada valor numérico en una determinada columna del dataset. En el eje X se representa la variable continua y en el eje Y la frecuencia de la misma.

Para generar este tipo de visualización utilizaremos `ggplot() + geom_histogram()`. Veamos por ejemplo como se distribuyen las superficies cubiertas (`surface_covered`) de las propiedades publicadas:

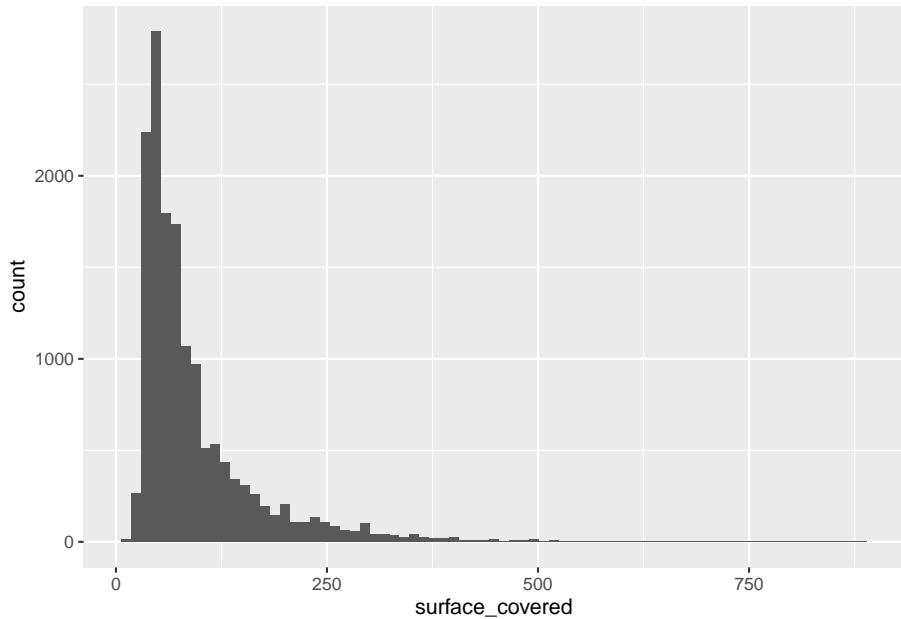
```
ggplot(datos_amba)+  
  geom_histogram(aes(x=surface_covered))
```



El eje X muestra la cantidad de m<sup>2</sup> cubiertos que tienen las propiedades publicadas y el eje Y la cantidad de veces que aparece cada superficie. Por lo tanto, podemos ver que hay muchas propiedades con “poca” superficie cubierta y pocas propiedades con “mucha” superficie cubierta.

Para facilitar la interpretación, podemos modificar el ancho de las barras (bins) que por defecto el valor es 30. Probemos con bins=75:

```
ggplot(datos_amba)+  
  geom_histogram(aes(x=surface_covered), bins=75)
```

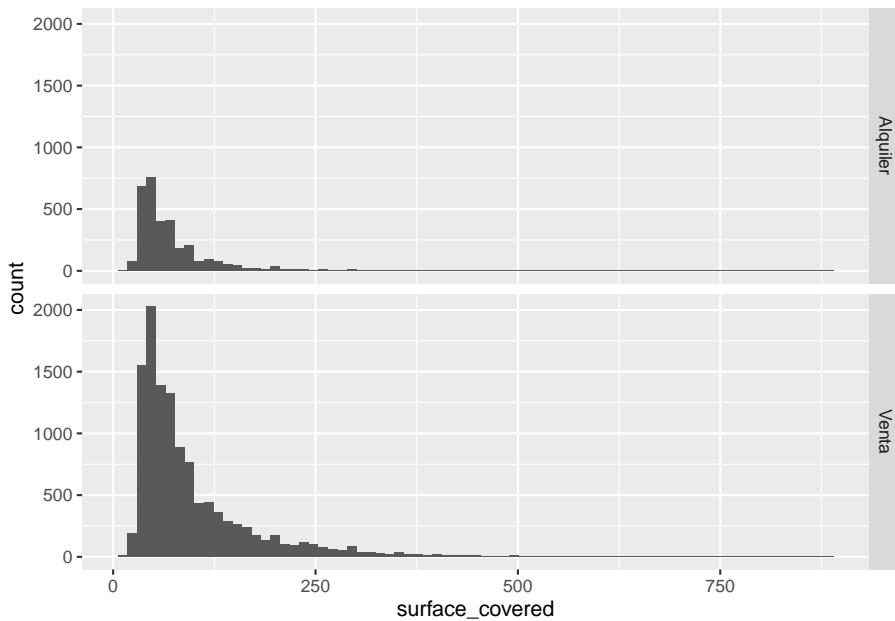


¿Notan el cambio? El eje Y disminuyó porque los conteos se agruparon en intervalos más pequeños sobre el eje X.

Podemos observar que hay más propiedades por debajo de los 100m<sup>2</sup> cubiertos que por encima, y que la mayor cantidad de observaciones se ubica alrededor de los 50m<sup>2</sup>. También podemos detectar algunos outliers que tienen alrededor de 800m<sup>2</sup>.

Pero desagreguemos aún más nuestros datos y sumemos una variable categórica (tipo de operación) que nos permita facetar/dividir el gráfico. Para esto utilizaremos `facet_grid()`:

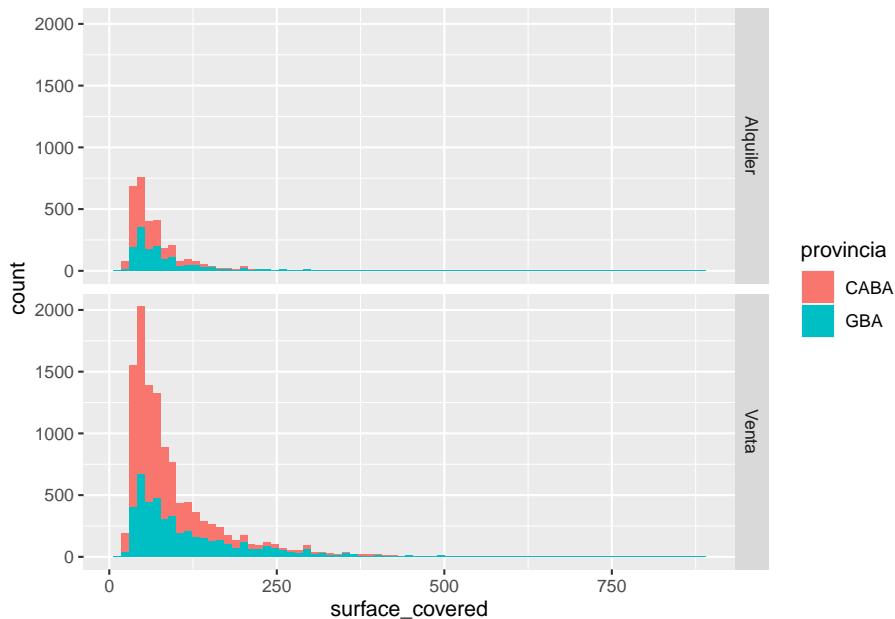
```
ggplot(datos_amba)+  
  geom_histogram(aes(x=surface_covered), bins=75) +  
  facet_grid(operation_type~.)
```



La tendencia se mantiene similar en ambos casos, con una mayoría de propiedades de aproximadamente 50m<sup>2</sup> cubiertos. Sin embargo, se detecta una gran diferencia en la cantidad de propiedades publicadas para cada tipo de operación.

Incorporemos una variable más (provincia) que nos permita comprender si los comportamientos de la variable `surface_covered` cambian entre CABA y PBA. En este caso la agregaremos como un color de relleno (fill):

```
ggplot(datos_amba)+  
  geom_histogram(aes(x=surface_covered, fill=provincia), bins=75)+  
  facet_grid(operation_type~.)
```



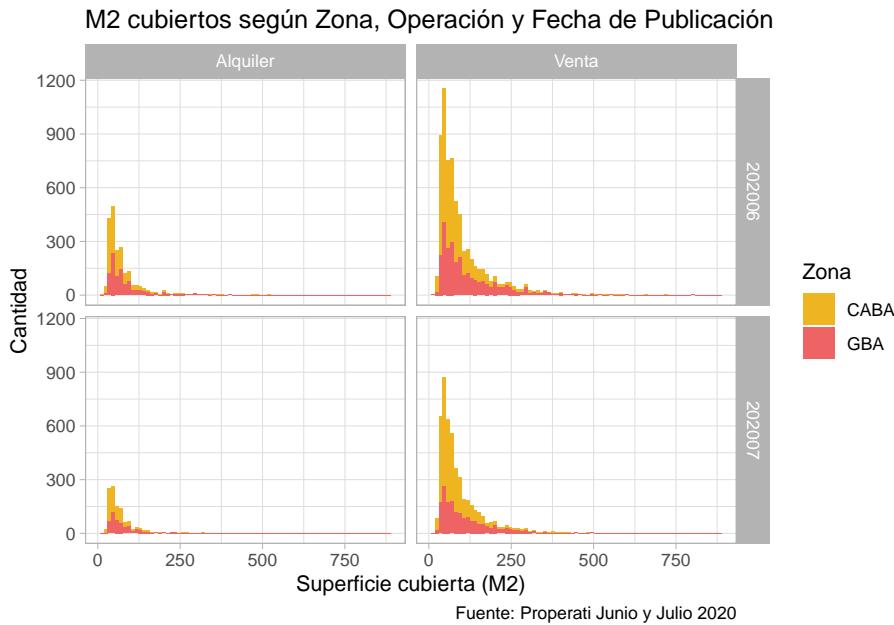
Podemos ver que tanto en CABA como en AMBA, los comportamientos de la variable analizada son similares.

Por último probemos sumar una variable más dentro del facetado (fecha de publicación), agreguemos etiquetas (labs) y elijamos los colores del gráfico (scale\_fill y theme).

Cabe destacar que hay 2 tipos de escalas de color:

- Las ya establecidas (listas para usar): Brewer, Viridis. Recomiendo utilizar estas que en `ggplot2` las van a encontrar como `scale_fill_viridis_c()`, `scale_fill_brewer()` y demás variantes.
- Las personalizadas, donde nosotros elegimos todos los colores: Gradiente (`scale_fill_gradient()`), Manual (`scale_fill_manual()`) *Para obtener un listado de colores pueden ver este link*

```
ggplot(datos_amba)+  
  geom_histogram(aes(x=surface_covered, fill=provincia), bins=75)+  
  facet_grid(created_on~operation_type)+  
  labs(title="M2 cubiertos según Zona, Operación y Fecha de Publicación",  
       fill="Zona",  
       x="Superficie cubierta (M2)",  
       y="Cantidad",  
       caption="Fuente: Properati Junio y Julio 2020") +  
  scale_fill_manual(values = c("goldenrod2", "indianred2"))+  
  theme_light()
```



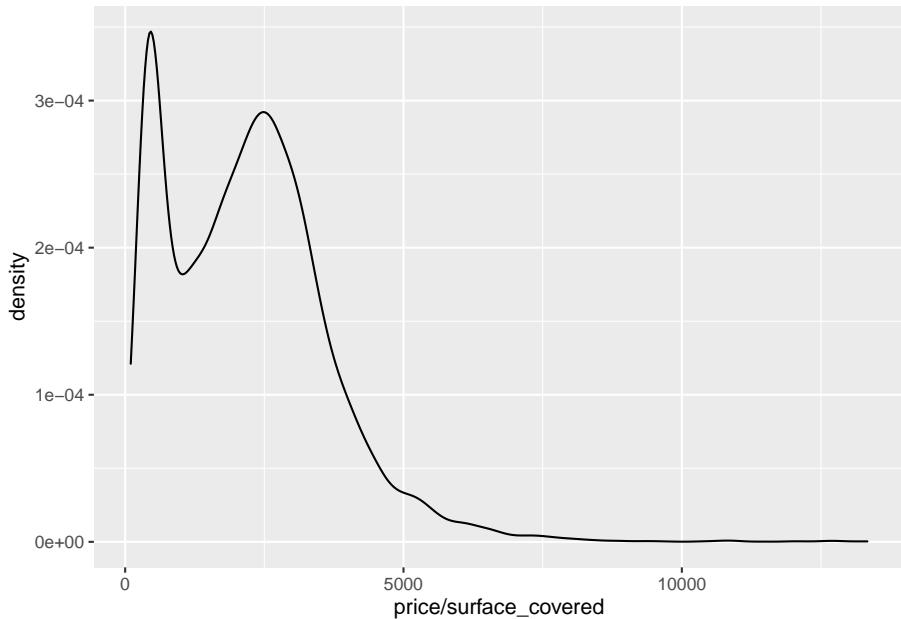
En el histograma anterior se puede ver que si bien en ambos meses las distribuciones se mantienen, en Junio hay mayor cantidad de publicaciones que en Julio.

### 3.1.2 Gráfico de Densidad

Este tipo de gráfico cumple la misma función que el Histograma, pero se caracteriza por generar una versión más “suavizada” en la que muestra la densidad de kernel a lo largo de toda la variable continua y no conteos por bins. Estos gráficos nos permiten ver cuales son los intervalos de la variable continua donde hay mayor probabilidad de encontrar registros.

Para generar esta visualización utilizaremos `ggplot() + geom_density()`. Veamos por ejemplo como se distribuyen los valores por m2 (price/surface\_covered) de las propiedades publicadas:

```
ggplot(datos_amba)+  
  geom_density(aes(x=price/surface_covered))
```

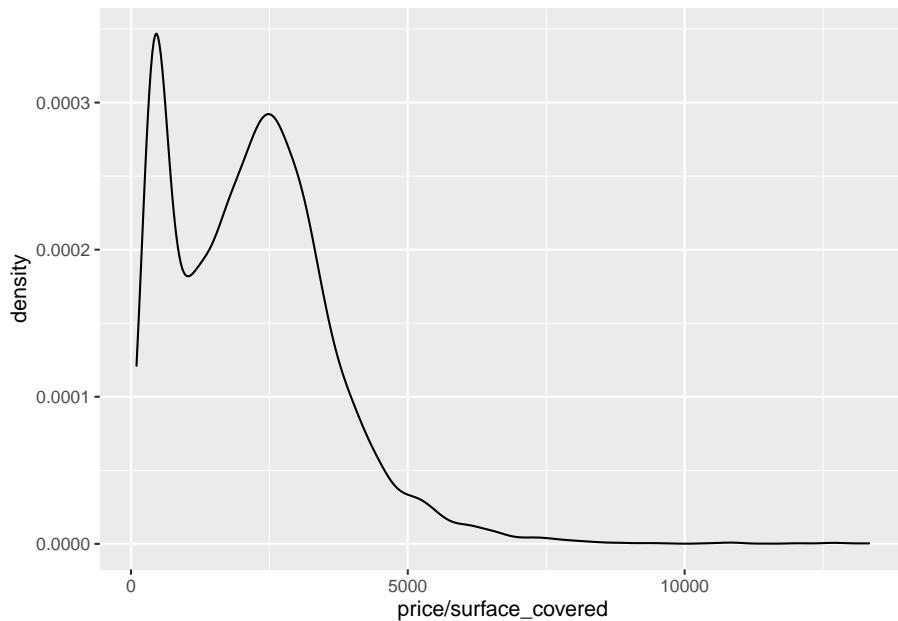


En primer lugar, tenemos los valores del eje Y en notación científica. Si queremos evitar esto es necesario que escribamos la siguiente línea de código:

```
options(scipen=999)
```

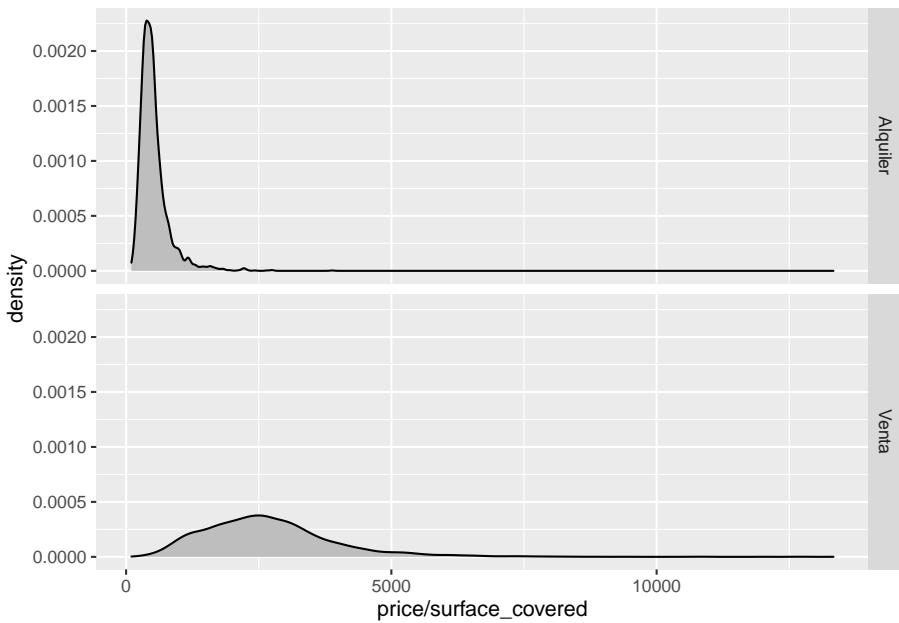
Y ahora si, ejecutar nuevamente el chunk anterior:

```
ggplot(datos_amba)+  
  geom_density(aes(x=price/surface_covered))
```



En el gráfico se ven 2 “picos” muy pronunciados que indican en qué partes del intervalo se concentran los valores. Esto podría deberse a que estamos trabajando con ambos tipos de operación (Alquiler y Venta), y como todos sabemos, los valores del m2 de ambos son muy diferentes, tanto por la moneda (ARS vs USD) como por los montos. Por lo tanto, es muy probable que cada uno de esos “picos” se corresponda al valor del m2 de cada una de las operaciones. Para poder desemascarar esto recurramos nuevamente al facetado:

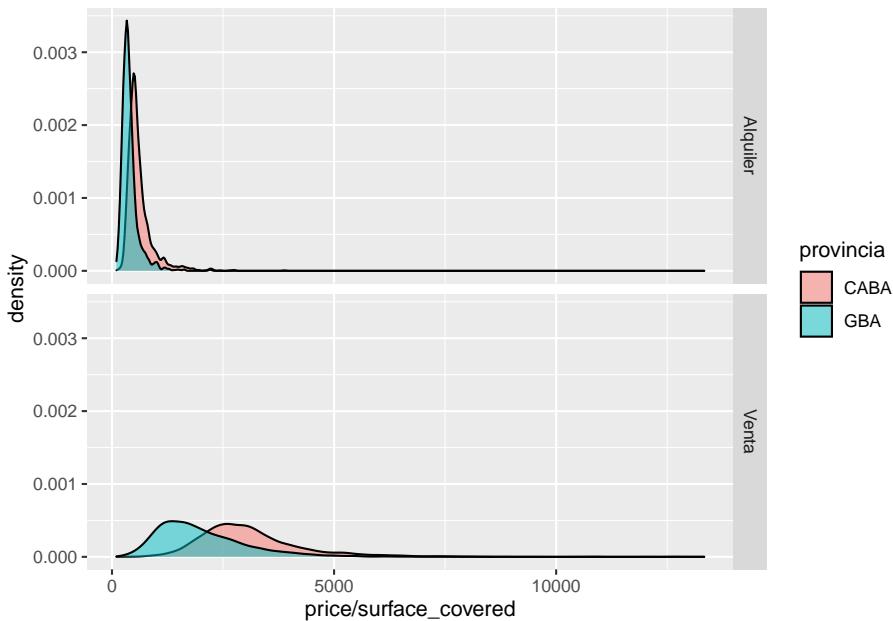
```
ggplot(datos_amba)+  
  geom_density(aes(x=price/surface_covered), fill="gray") +  
  facet_grid(operation_type~.)
```



Las operaciones tienen una distribución del valor del m<sup>2</sup> muy diferentes entre sí. Se ve muy claro que los alquileres tienen un rango de precios que va desde 300 a 500\$ARS aprox mientras que las ventas distribuyen sus publicaciones a lo largo de un rango bastante mayor entre 1.500USD y 3.500USD aproximadamente.

Pero como los precios también deberían variar bastante de acuerdo a la zona geográfica, probemos agregar una variable categórica más que nos ayude a diferenciar CABA de GBA:

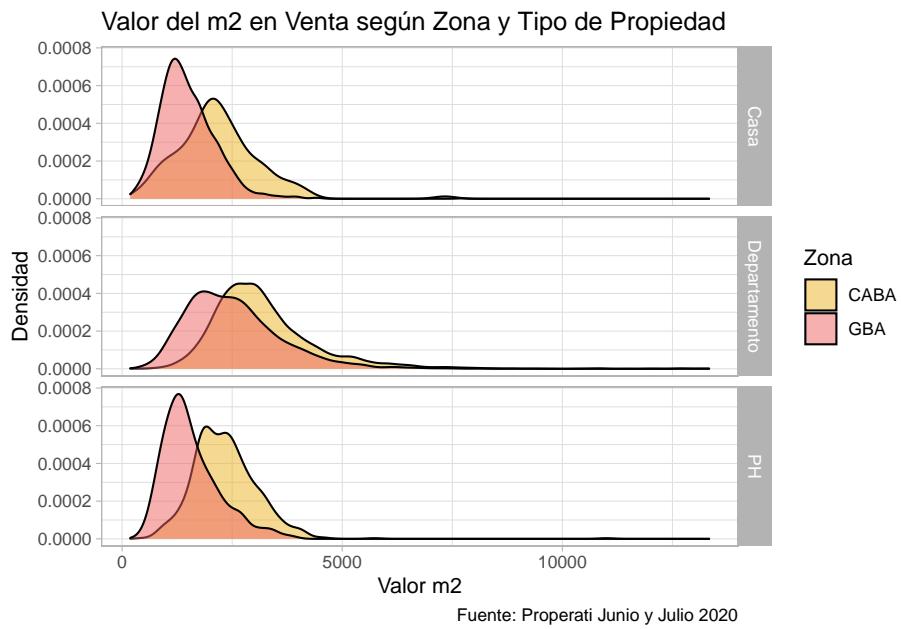
```
ggplot(datos_amba)+  
  geom_density(aes(x=price/surface_covered, fill=provincia), alpha=0.5) +  
  facet_grid(operation_type~.)
```



Obviamente para cada partido, localidad y calle específica los valores van a variar, pero a simple vista podríamos notar que en ambos casos el valor del m<sup>2</sup> de la mayoría de propiedades es mayor en CABA.

Ahora analicemos como se comporta el valor del m<sup>2</sup> de venta según el tipo de propiedad en CABA y GBA y aprovechemos para agregar etiquetas y colores a nuestro gráfico:

```
ggplot(datos_amba %>%
         filter(operation_type=="Venta"))+
  geom_density(aes(x=price/surface_covered, fill=provincia), alpha=0.5) +
  facet_grid(property_type~.)+
  labs(title="Valor del m2 en Venta según Zona y Tipo de Propiedad",
       fill="Zona",
       x="Valor m2",
       y="Densidad",
       caption="Fuente: Properati Junio y Julio 2020")+
  scale_fill_manual(values = c("goldenrod2", "indianred2"))+
  theme_light()
```

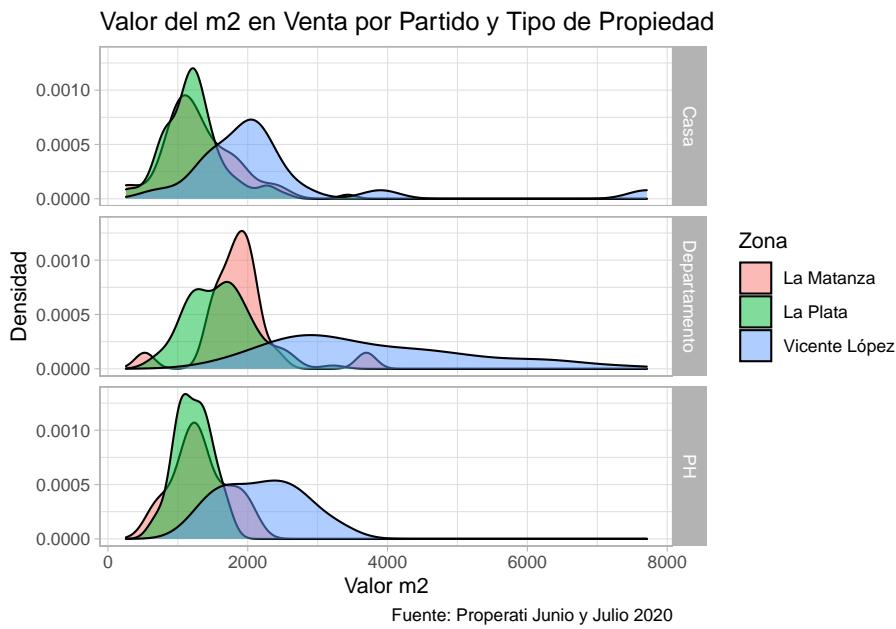


En las 3 tipologías de vivienda los valores del m<sup>2</sup> que predominan en CABA son mayores a los de GBA.

Y ahora elijamos 3 partidos para comparar:

```
ggplot(datos_amba %>%
      filter(operation_type=="Venta" & partido==c("Vicente López", "La Matanza", "La"))
      geom_density(aes(x=price/surface_covered, fill=partido), alpha=0.5) +
      facet_grid(property_type~.)+
      labs(title="Valor del m2 en Venta por Partido y Tipo de Propiedad",
           fill="Zona",
           x="Valor m2",
           y="Densidad",
           caption="Fuente: Properati Junio y Julio 2020")+
      theme_light()
```

### 3.2. DISTRIBUCIÓN DE VALORES CONTINUOS ASOCIADOS A UNA VARIABLE CATEGÓRICA: GRÁFICO

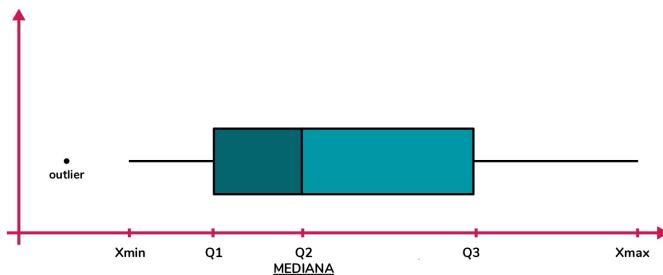


En las 3 tipologías se ve que Vicente López tiene los precios más elevados por m2, y a su vez que los departamentos de La Matanza superan en precio a los de La Plata. Sin embargo, para casas y PHs, La Matanza y La Plata tienen precios muy similares.

## 3.2 Distribución de valores continuos asociados a una variable categórica: Gráfico de Cajas

Un Gráfico de Cajas sirve para comparar la distribución y tendencia central de varias categorías de una variable.

Veamos como luce un gráfico de este tipo:



Para cumplir con el objetivo de analizar la distribución de una determinada muestra de datos, **la visualización distribuye los datos en cuartiles: Q1, Q2 y Q3**. Cabe destacar que, los cuartiles son los valores que dividen a la muestra (cantidad de registros) en 4 partes iguales.

**Q1:** El primer cuartil está representado por el mayor valor incluido en el 1/4 más bajo. Es decir que, el 25% de la muestra de datos es menor que este valor.

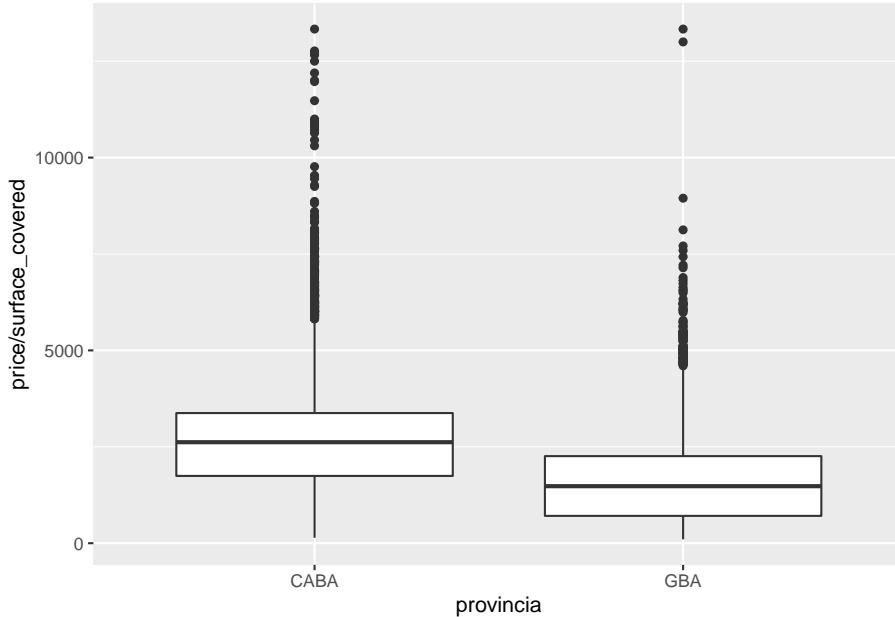
**Q2:** El segundo cuartil está representado por el mayor valor incluido en el 2/4. Es decir que, el 50% de la muestra de datos es menor que este valor. Este valor indica la **mediana de la serie**.

**Q3:** El tercer cuartil está representado por el mayor valor incluido en el 3/4. Es decir que, el 75% de la muestra de datos es menor que este valor.

En este gráfico también pueden aparecer **outliers**, es decir valores extremos que haya dentro de los conjuntos de datos.

Para generar esta visualización utilizaremos `ggplot() + geom_boxplot()`. Veamos por ejemplo como se distribuyen los valores por m2 (price/surface\_covered) por zona:

```
ggplot(datos_amba) +
  geom_boxplot(aes(x = provincia, y = price/surface_covered))
```

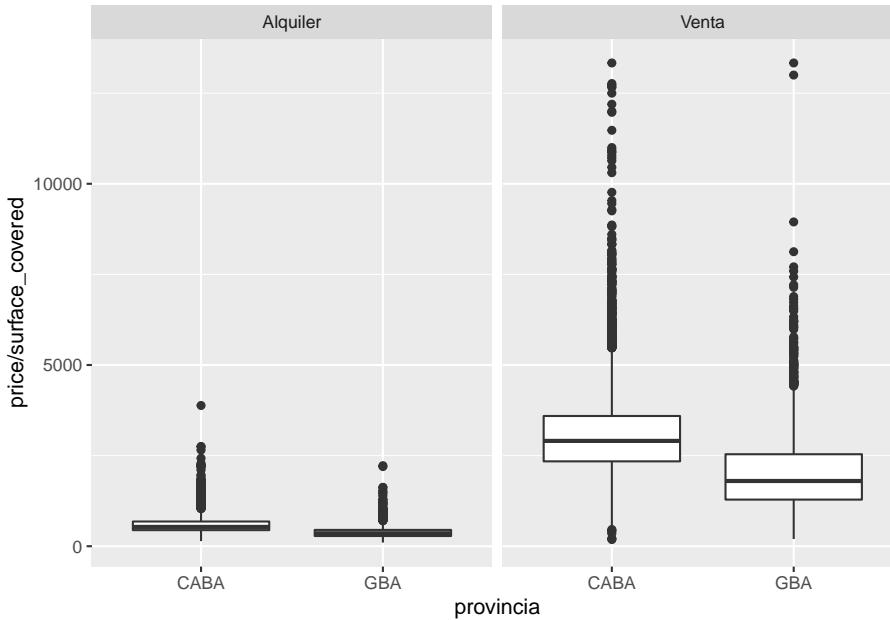


En el gráfico anterior podemos observar que la mediana del valor del m2 en CABA es superior al de GBA, pero estamos mezclando datos de ambos tipos

### 3.2. DISTRIBUCIÓN DE VALORES CONTINUOS ASOCIADOS A UNA VARIABLE CATEGÓRICA: GRÁFICO

de operaciones (alquileres y ventas), lo cual no es correcto. Para mejorar este aspecto facetemos el gráfico según tipo de operación:

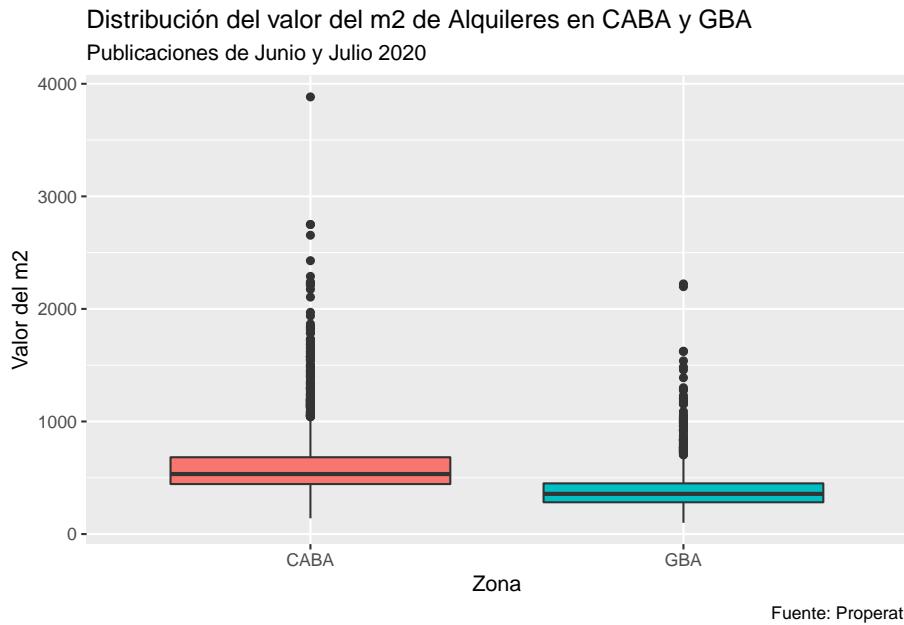
```
ggplot(datos_amba) +
  geom_boxplot(aes(x = provincia, y = price/surface_covered)) +
  facet_grid(~operation_type)
```



Si bien los valores que se manejan en cada operación son muy diferentes entre sí, en ambos casos se mantiene que la **mediana del valor del m<sup>2</sup> en CABA es superior que en GBA**. Sin embargo, como las 2 escalas son muy diferentes, casi no se ven las cajas correspondientes a los alquileres.

Hagamos un filtro por Alquileres para ver mejor que ocurre ahí:

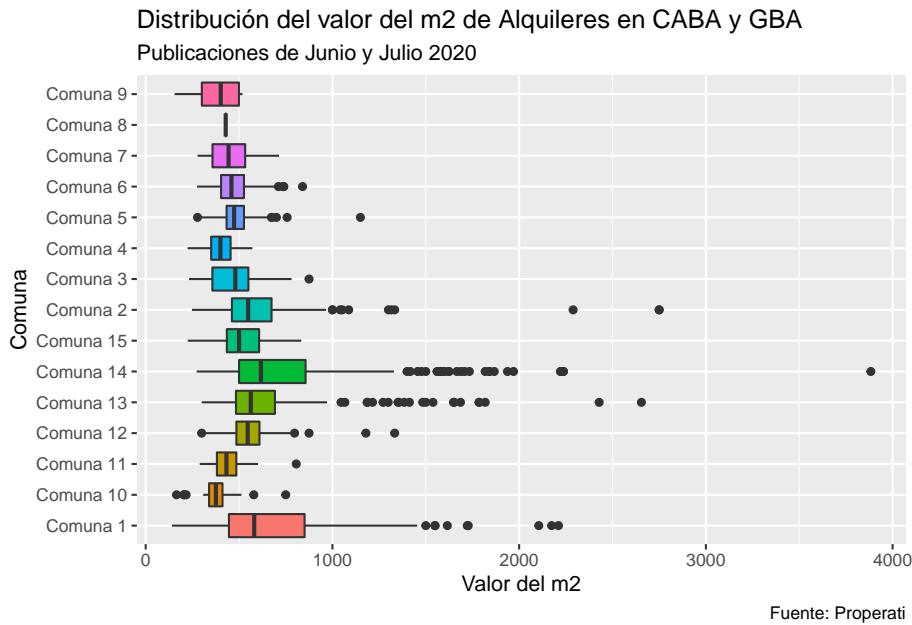
```
ggplot(datos_amba %>%
         filter(operation_type=="Alquiler")) +
  geom_boxplot(aes(x = provincia, y = price/surface_covered, fill=provincia), show.legend = FALSE)
  labs(title = "Distribución del valor del m2 de Alquileres en CABA y GBA",
       subtitle = "Publicaciones de Junio y Julio 2020",
       y = "Valor del m2",
       x = "Zona",
       caption = "Fuente: Properati")
```



La mediana de CABA está por encima pero claramente dentro de CABA hay muchas diferencias entre Comunas. Grafiquemos esto con datos de CABA desagregados por Comuna:

```
ggplot(datos_amba %>%
         filter(operation_type=="Alquiler" & provincia=="CABA")) +
  geom_boxplot(aes(x = partido, y = price/surface_covered, fill=partido), show.legend = TRUE) +
  labs(title = "Distribución del valor del m2 de Alquileres en CABA y GBA",
       subtitle = "Publicaciones de Junio y Julio 2020",
       y = "Valor del m2",
       x = "Comuna",
       caption = "Fuente: Properati")+
  coord_flip()
```

### 3.2. DISTRIBUCIÓN DE VALORES CONTINUOS ASOCIADOS A UNA VARIABLE CATEGÓRICA: GRÁFICO

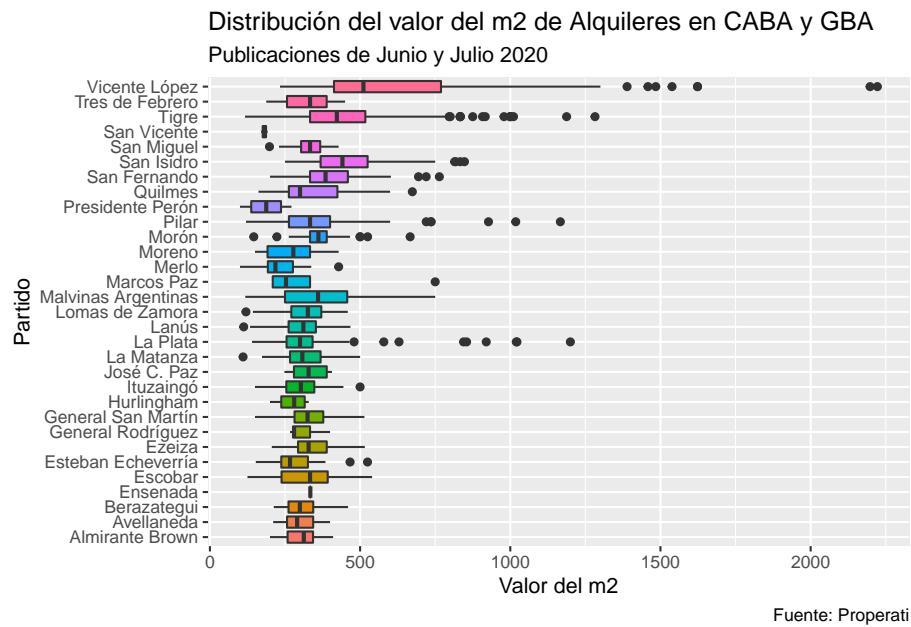


Se ve que:

- La Comuna 14 y la Comuna 1 tienen una mediana muy similar en el valor del m<sup>2</sup> (el más alto de CABA).
- La Comuna 14 presenta la mayor cantidad de outliers, motivo por el cual al calcular promedios siempre está por encima de la Comuna 1.
- La Comuna 1 presenta la mayor variación en el valor del m<sup>2</sup>: va desde 200ARS hasta 1500ARS aprox. Esto se puede deber a la heterogeneidad de barrios que contiene.
- La Comuna 8 tiene muy pocas observaciones.
- La Comuna 10 presenta el menor valor del m<sup>2</sup>.

Veamos que pasa si filtramos solo PBA y lo desagregamos por Partido:

```
ggplot(datos_amba %>%
      filter(operation_type=="Alquiler" & provincia=="GBA")) +
  geom_boxplot(aes(x = partido, y = price/surface_covered, fill=partido), show.legend = FALSE) +
  labs(title = "Distribución del valor del m2 de Alquileres en CABA y GBA",
       subtitle = "Publicaciones de Junio y Julio 2020",
       y = "Valor del m2",
       x = "Partido",
       caption = "Fuente: Properati")+
  coord_flip()
```



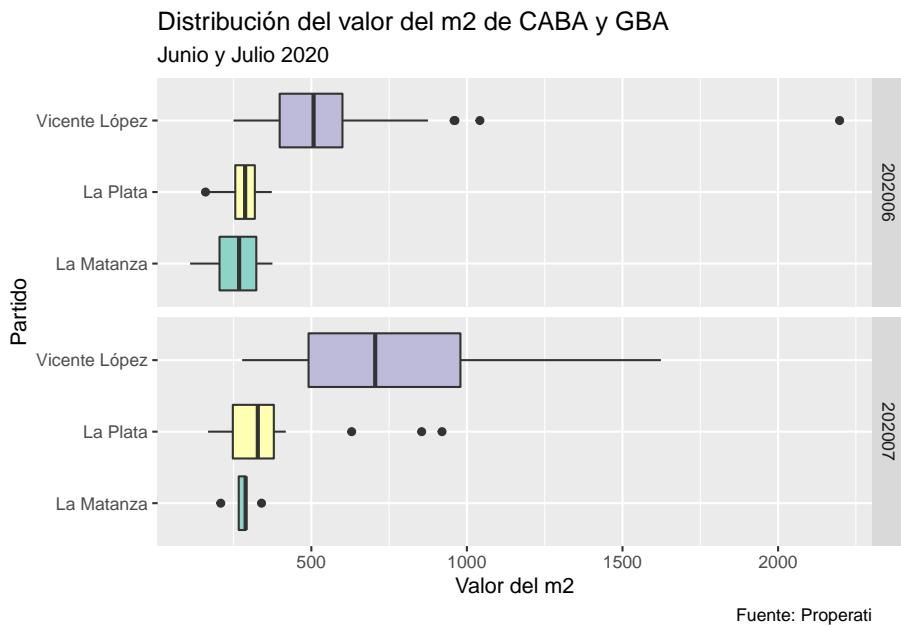
Acá vemos que:

- Vicente López presenta la mayor variación en el valor del m<sup>2</sup> y la mediana más alta de los partidos incluidos en la base.
- Presidente Perón es el partido de la base con menor valor del m<sup>2</sup>.

Este análisis también podríamos hacerlo comparando entre partidos específicos y diferenciando por mes de publicación, por ejemplo:

```
ggplot(datos_amba %>%
  filter(operation_type=="Alquiler" & partido==c("Vicente López", "La Matanza",
  geom_boxplot(aes(x = partido, y = price/surface_covered, fill=partido), show.legend =
  labs(title = "Distribución del valor del m2 de CABA y GBA",
  subtitle = "Junio y Julio 2020",
  y = "Valor del m2",
  x = "Partido",
  color = "Partido",
  caption = "Fuente: Properati")+
  facet_grid(created_on~.)+
  scale_fill_brewer(palette = "Set3")+
  coord_flip()
```

### 3.3. RELACIÓN ENTRE VARIABLES NUMÉRICAS: GRÁFICO DE DISPERSIÓN 69

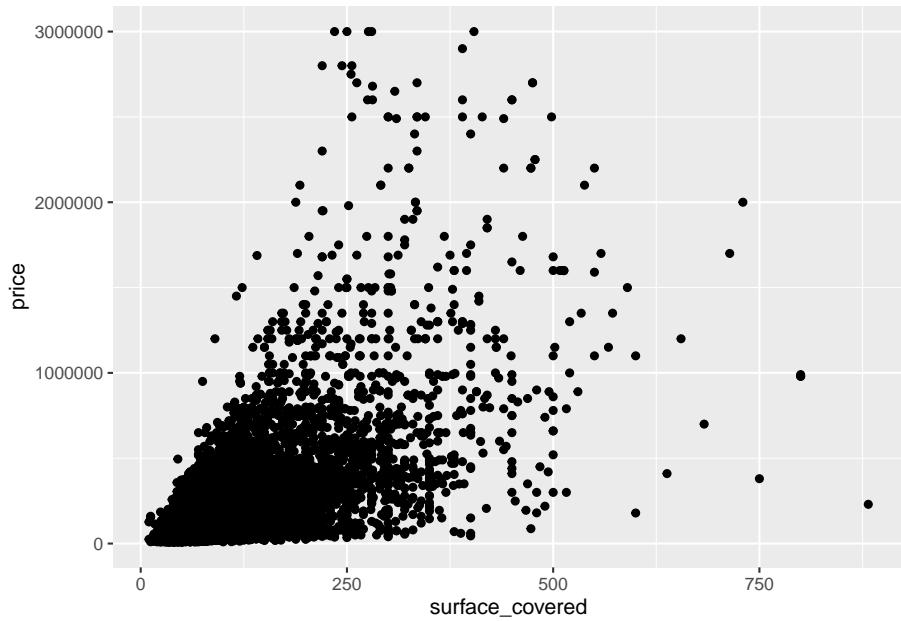


### 3.3 Relación entre variables numéricas: Gráfico de Dispersion

Ahora veamos el clásico gráfico de puntos o scatter plot que muestra la dispersión que existe entre 2 variables numéricas representadas en los 2 ejes X e Y, y que permite identificar si existe o no una relación entre ambas.

Veamos por ejemplo, si existe relación entre la superficie cubierta (surface\_covered) y el precio total (price) de las propiedades:

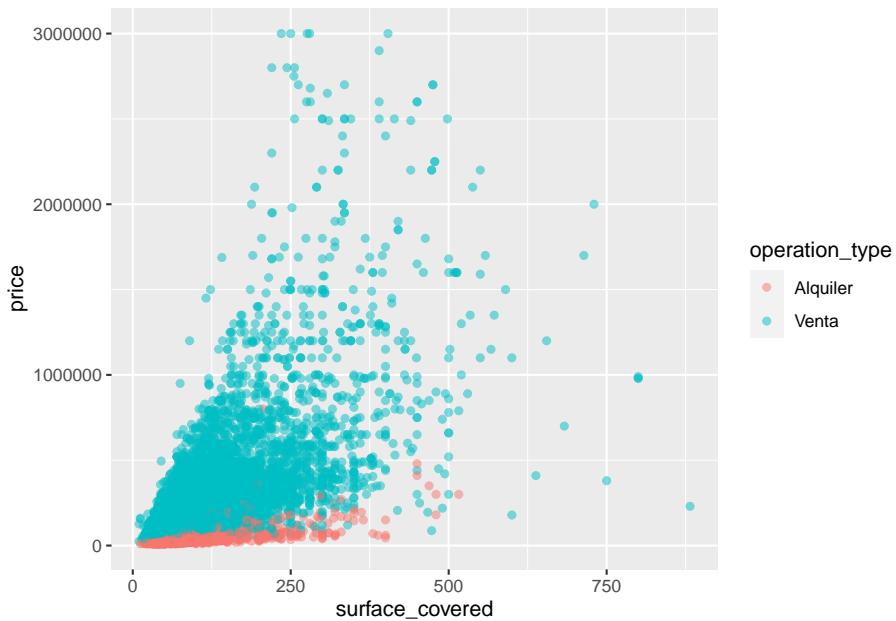
```
ggplot(datos_amba)+  
  geom_point(aes(x=surface_covered, y=price))
```



Tiene sentido, a mayor superficie cubierta, mayor valor. Pero veamos esto desagregado entre tipos de operaciones a ver que pasa:

```
ggplot(datos_amba)+  
  geom_point(aes(x=surface_covered, y=price, color=operation_type), alpha=0.5)
```

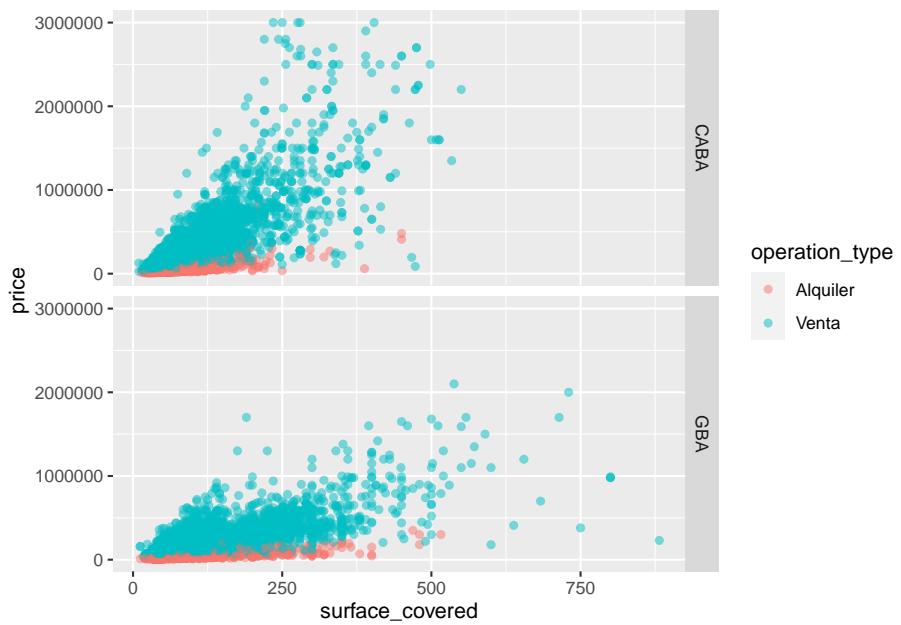
### 3.3. RELACIÓN ENTRE VARIABLES NUMÉRICAS: GRÁFICO DE DISPERSIÓN 71



A los gráficos de dispersión también se les puede agregar una tercera variable categórica o numérica que puede verse reflejada en la estética de los puntos (color, forma o tamaño).

Agreguemos una tercer variable (categórica) a nuestro gráfico: la provincia

```
ggplot(datos_amba)+  
  geom_point(aes(x=surface_covered, y=price, color=operation_type), alpha=0.5) +  
  facet_grid(provincia~.)
```

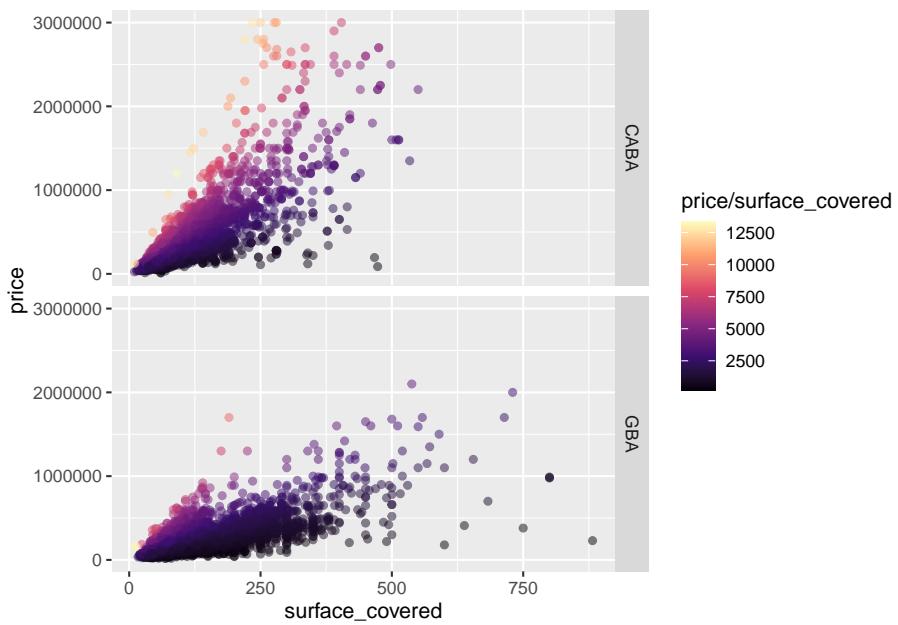


Podemos ver que en GBA no crece tan rápido la relación como en CABA.

Quedémonos solo con las ventas y agreguemos una tercer variable numérica: el valor del m<sup>2</sup>

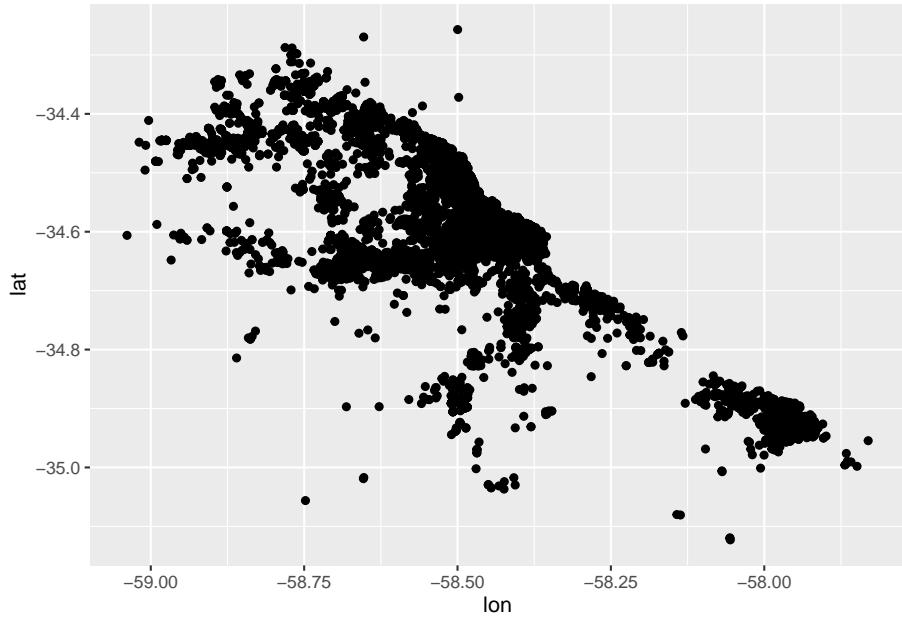
```
ggplot(datos_amba %>%
         filter(operation_type=="Venta"))+
  geom_point(aes(x=surface_covered, y=price, color=price/surface_covered), alpha=0.5) +
  facet_grid(provincia~.) +
  scale_color_viridis_c(option="magma")
```

### 3.3. RELACIÓN ENTRE VARIABLES NUMÉRICAS: GRÁFICO DE DISPERSIÓN73



Ahora aprovechamos que tenemos longitud y latitud de cada registro y hagamos un gráfico de puntos que nos permita ver la relación entre ambas variables:

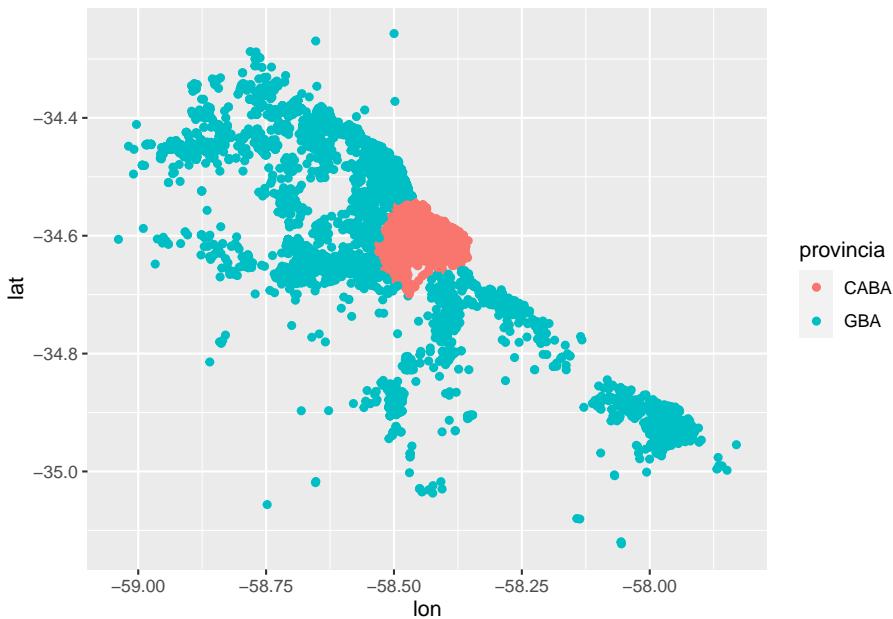
```
ggplot()+
  geom_point(data=datos_amba, aes(x=lon, y=lat))
```



¿A qué se parece? ¿Tiene forma de AMBA? Probemos coloreando según la variable provincia.

```
ggplot()+
  geom_point(data=datos_amba, aes(x=lon, y=lat, color=provincia))
```

### 3.4. RELACIÓN ENTRE VARIABLES CATEGÓRICAS: GRÁFICO DE MATRIZ75



## 3.4 Relación entre variables categóricas: Gráfico de Matriz

Este tipo de gráfico representa un mapa de calor bidimensional que muestra la frecuencia que existe entre 2 variables dentro de la base de datos.

Para poder desarrollar la visualización es necesario elegir 3 variables:

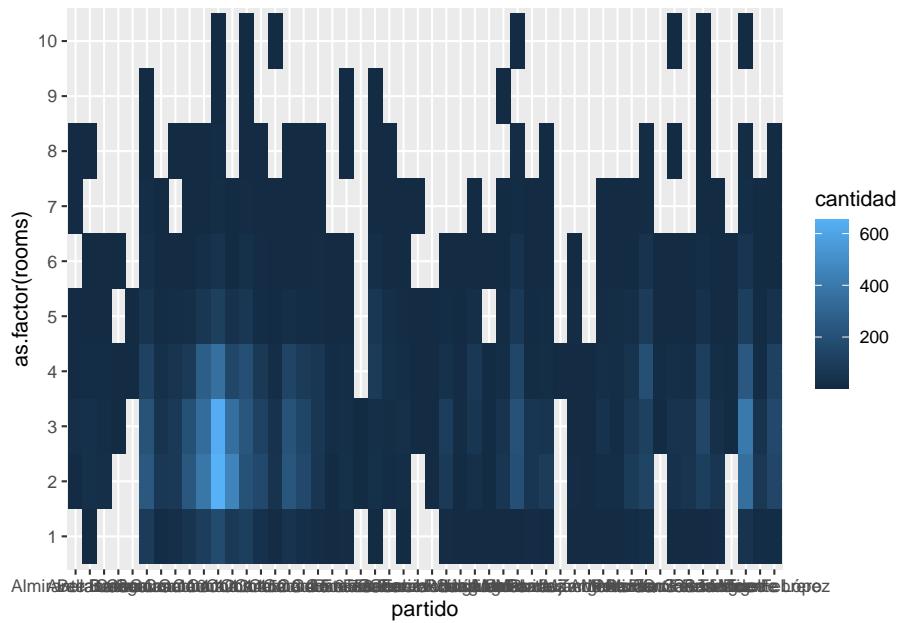
**x**: variable a ubicar en el eje X

**y**: variable a ubicar en el eje Y

**fill**: valor numérico que será representado a partir de los colores

Veamos un ejemplo relacionando el partido y la cantidad de ambientes:

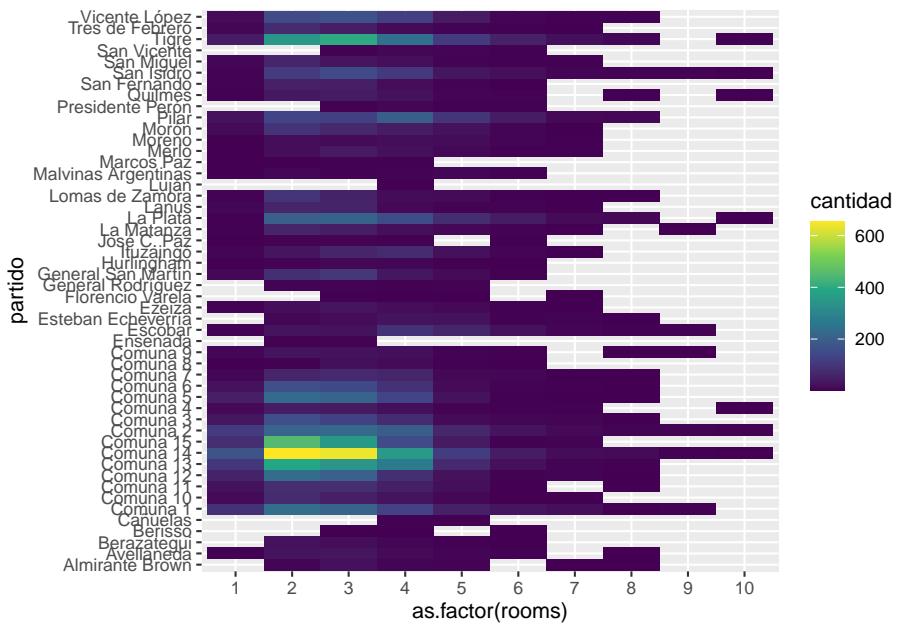
```
ggplot(datos_amba %>%
         group_by(partido, rooms) %>%
         summarise(cantidad=n())) +
  geom_tile(aes(x = partido,
                y = as.factor(rooms),
                fill = cantidad))
```



Mejoremos con `coord_flip()` el aspecto de la visualización:

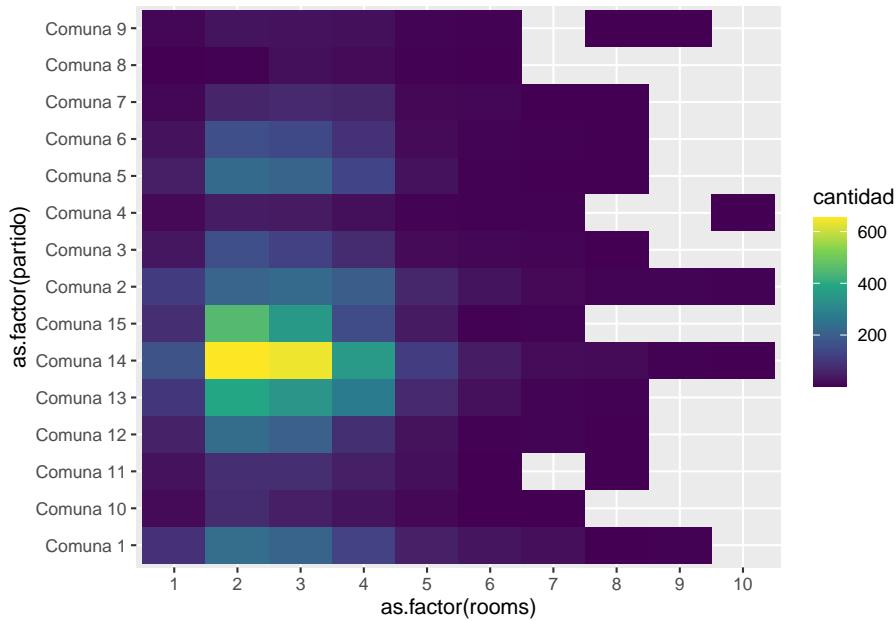
```
ggplot(datos_amba %>%
         group_by(partido, rooms) %>%
         summarise(cantidad=n())) +
  geom_tile(aes(x = partido,
                y = as.factor(rooms),
                fill = cantidad)) +
  scale_fill_viridis_c() +
  coord_flip()
```

### 3.4. RELACIÓN ENTRE VARIABLES CATEGÓRICAS: GRÁFICO DE MATRIZ77



Se puede ver por ejemplo que la relación Comuna 14 + 2 ambientes, y Comuna 14 + 3 ambientes es la que más se repite (más de 600 veces cada una). Filtremos solo CABA:

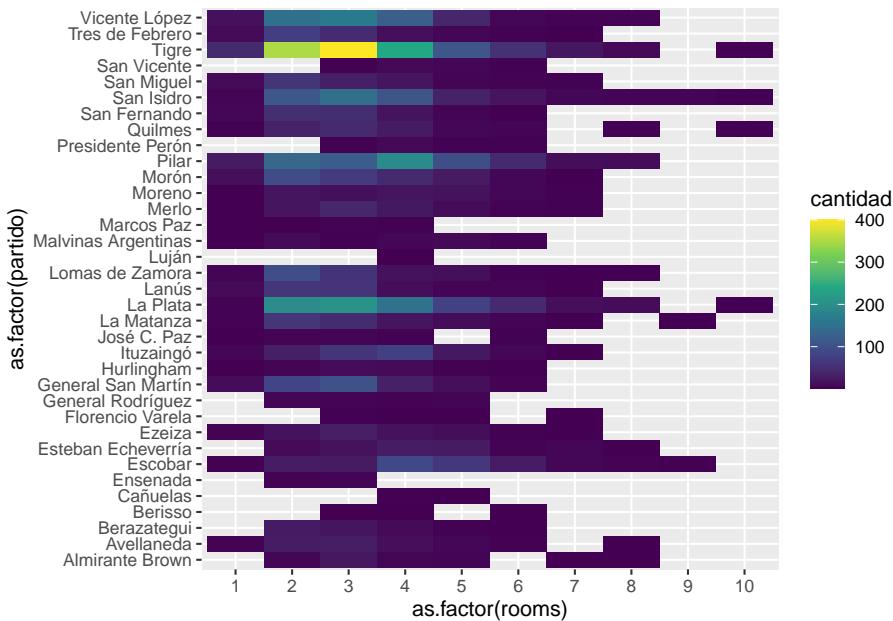
```
ggplot(datos_amba %>%
  filter(provincia=="CABA") %>%
  group_by(partido, rooms) %>%
  summarise(cantidad=n())) +
  geom_tile(aes(x = as.factor(partido),
                y = as.factor(rooms),
                fill = cantidad)) +
  scale_fill_viridis_c() +
  coord_flip()
```



Obviamente se sigue viendo lo mismo de la Comuna 14 pero también se empieza a ver como la Comuna 13 y 15 también aparecen mucho con propiedades de 2 y 3 ambientes. Veamos el caso de GBA:

```
ggplot(datos_amba %>%
          filter(provincia=="GBA") %>%
          group_by(partido, rooms) %>%
          summarise(cantidad=n())) +
  geom_tile(aes(x = as.factor(partido),
                y = as.factor(rooms),
                fill = cantidad)) +
  scale_fill_viridis_c() +
  coord_flip()
```

### 3.5. RELACIÓN ENTRE VARIABLE NUMÉRICA Y CATEGÓRICA: GRÁFICO DE BARRAS79



En GBA la mayor cantidad se ve en Tigre con 3 ambientes, seguida por 2 y 4. También hay varias propiedades de 2 y 3 ambientes en La Plata.

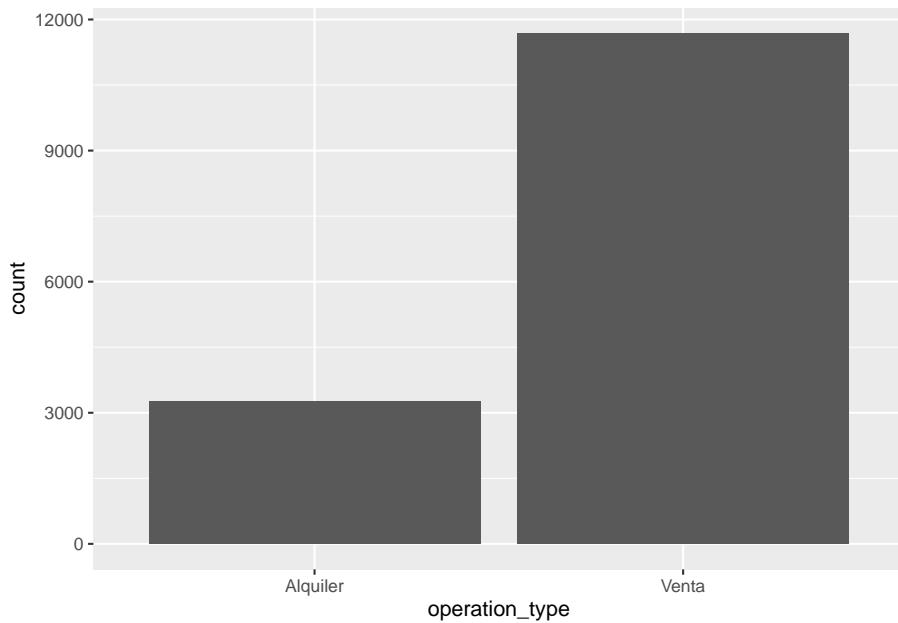
### 3.5 Relación entre variable numérica y categórica: Gráfico de Barras

El gráfico de barras representa, a partir de la longitud de las barras, el valor numérico (eje Y) asociado a cada entidad de la variable categórica (eje X).

Al igual que en el resto de visualizaciones, es necesario elegir una variable para el eje X y otra para el eje Y (acá se llama weight). Sin embargo, si no asignamos ninguna variable numérica a weight, el gráfico automáticamente va a calcular cuantas veces aparece cada categoría en la base de datos.

Por ejemplo, veamos cuantas observaciones hay por tipo de operación:

```
ggplot(datos_amba)+  
  geom_bar(aes(x=operation_type))
```

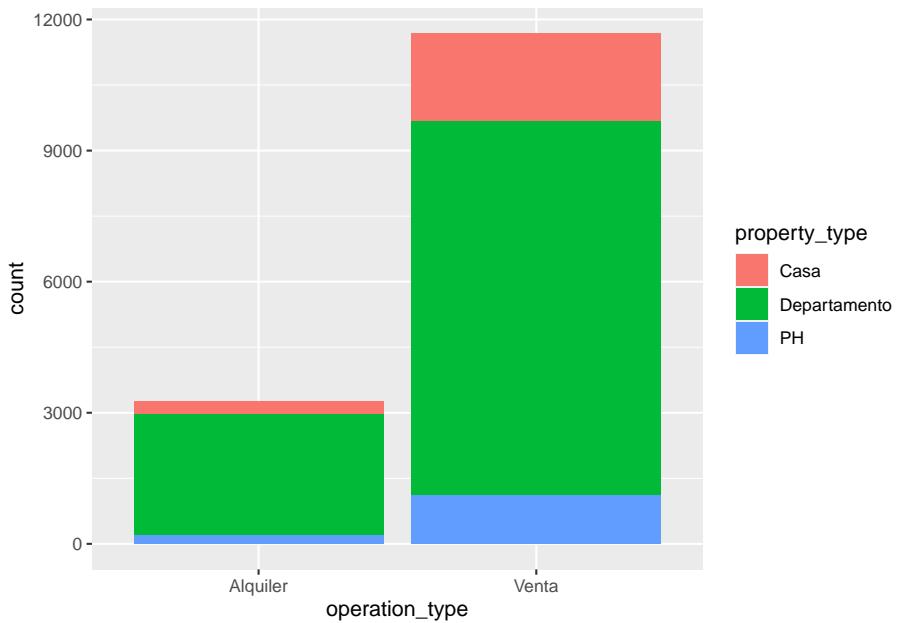


En el gráfico anterior podemos ver que hay alrededor de 12.000 propiedades en venta y 3.000 en alquiler.

Veamos esto desagregado por tipo de propiedad:

```
ggplot(datos_amba)+  
  geom_bar(aes(x=operation_type, fill=property_type))
```

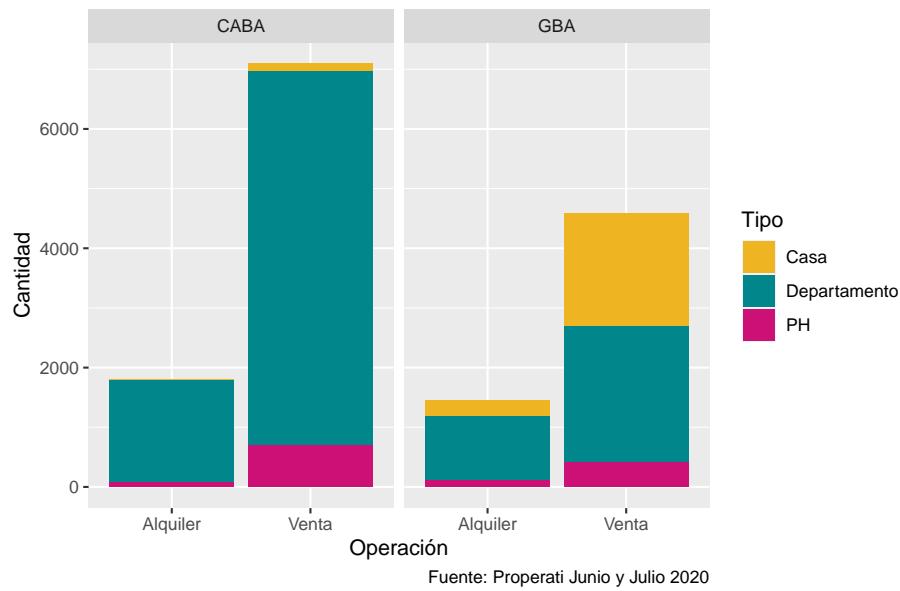
### 3.5. RELACIÓN ENTRE VARIABLE NUMÉRICA Y CATEGÓRICA: GRÁFICO DE BARRAS81



Como ya venimos viendo a lo largo de todo el capítulo, predominan los departamentos en ambos casos. Veamos esto también por zona y ajustemos cuestiones estéticas del gráfico:

```
ggplot(datos_amba)+  
  geom_bar(aes(x=operation_type, fill=property_type)) +  
  scale_fill_manual(values = c("goldenrod2", "turquoise4", "deeppink3")) +  
  facet_grid(~provincia)+  
  labs(title="Oferta publicada según Zona por Operación y Tipo de Propiedad",  
       fill="Tipo",  
       x="Operación",  
       y="Cantidad",  
       caption="Fuente: Properati Junio y Julio 2020")
```

Oferta publicada según Zona por Operación y Tipo de Propiedad



Siguen predominando los departamentos pero también se puede ver como la cantidad de casas en venta en GBA es muy similar a la de departamentos.

Las **barras apiladas con valores absolutos** es una de las opciones a la hora de graficar y la que nos hace por defecto ggplot2, pero existen 2 más: las barras apiladas con valores relativos (%) y las barras agrupadas con valores absolutos.

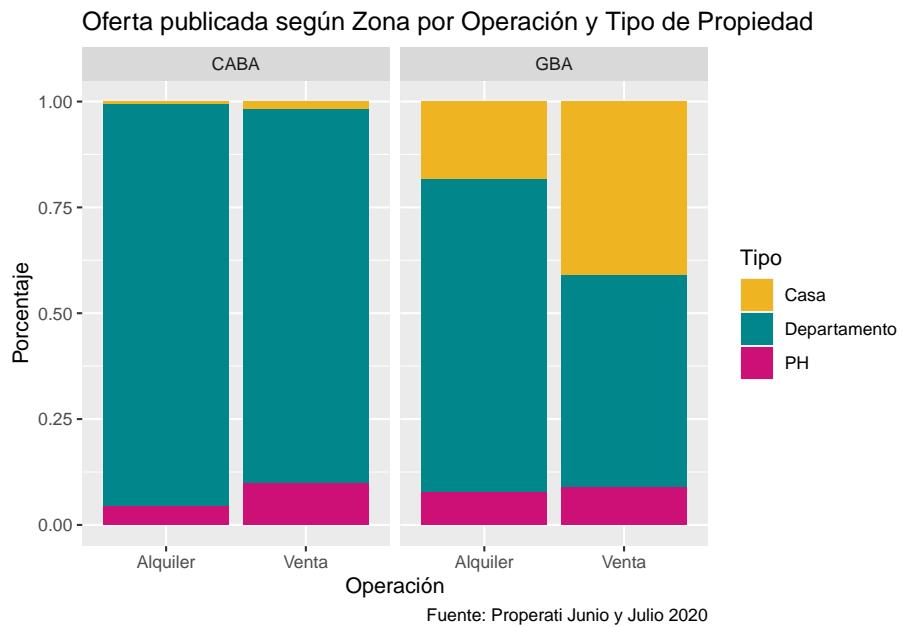
Veamos de que se trata cada una!

#### Barras apiladas con valores relativos (%)

Este tipo de gráfico ayuda a detectar las diferencias relativas que existen entre los valores continuos de cada grupo/categoría. Cada barra del gráfico muestra el total de cada categoría y se representa por el apilado de los porcentajes de cada valor. Con ggplot() usaremos position=position\_fill():

```
ggplot(datos_amba)+  
  geom_bar(aes(x=operation_type, fill=property_type), position=position_fill()) +  
  scale_fill_manual(values = c("goldenrod2", "turquoise4", "deeppink3")) +  
  facet_grid(~provincia)+  
  labs(title="Oferta publicada según Zona por Operación y Tipo de Propiedad",  
       fill="Tipo",  
       x="Operación",  
       y="Porcentaje",  
       caption="Fuente: Properati Junio y Julio 2020")
```

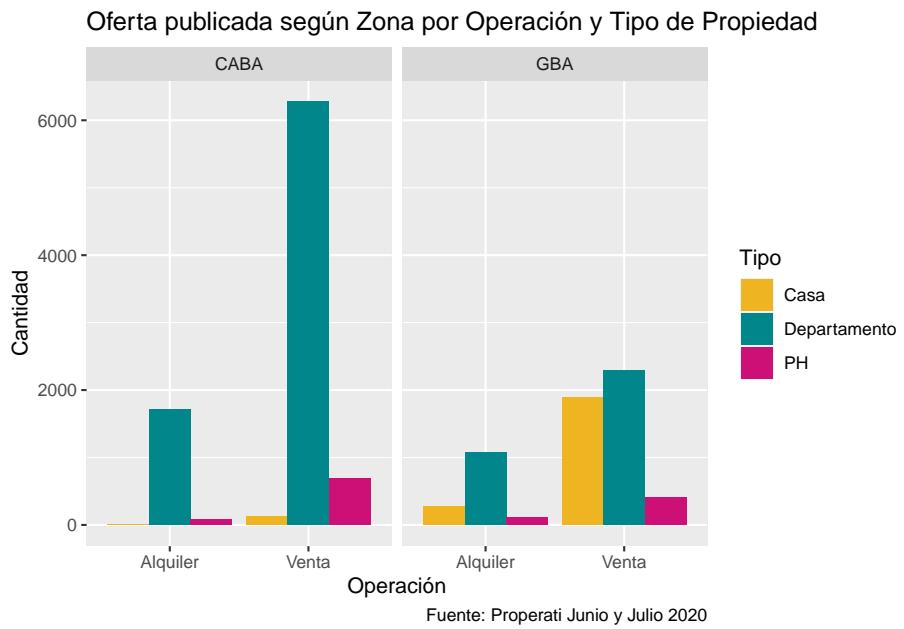
### 3.5. RELACIÓN ENTRE VARIABLE NUMÉRICA Y CATEGÓRICA: GRÁFICO DE BARRAS83



#### Barras agrupadas con valores absolutos

Este tipo de gráfico se utiliza cuando los datos absolutos se agrupan en 2 o más categorías dentro del mismo eje. Con `ggplot()` usaremos `position=position_dodge()`:

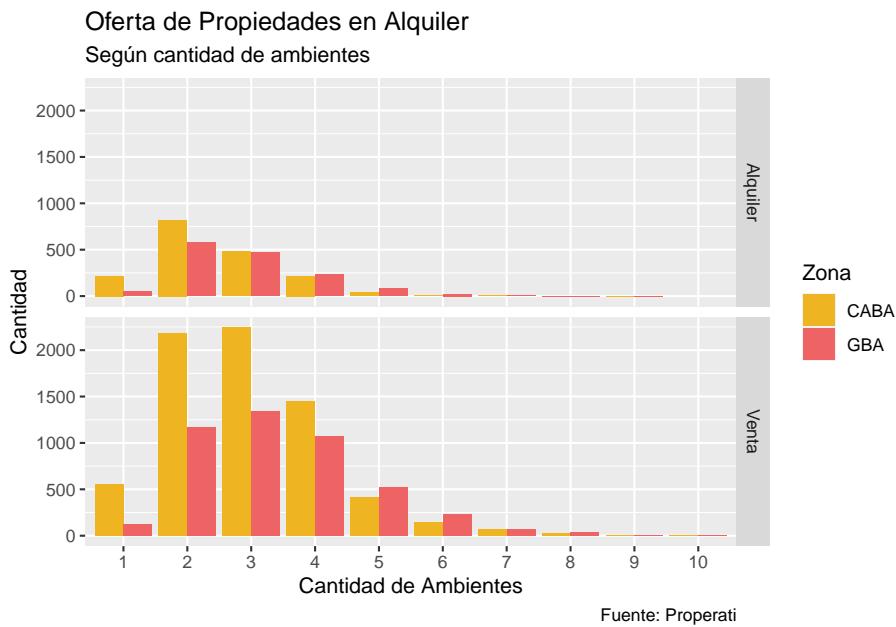
```
ggplot(datos_amba)+  
  geom_bar(aes(x=operation_type, fill=property_type), position=position_dodge()) +  
  scale_fill_manual(values = c("goldenrod2", "turquoise4", "deeppink3")) +  
  facet_grid(~provincia)+  
  labs(title="Oferta publicada según Zona por Operación y Tipo de Propiedad",  
       fill="Tipo",  
       x="Operación",  
       y="Cantidad",  
       caption="Fuente: Properati Junio y Julio 2020")
```



Ahora investiguemos otra variable de nuestro dataset, veamos **de cuántos ambientes son las propiedades que predominan en cada zona para cada operación inmobiliaria**:

```
ggplot(datos_amba)+  
  geom_bar(aes(x=as.factor(rooms), fill=provincia), position=position_dodge()) +  
  labs(title="Oferta de Propiedades en Alquiler",  
       subtitle="Según cantidad de ambientes",  
       fill="Zona",  
       x="Cantidad de Ambientes",  
       y="Cantidad",  
       caption="Fuente: Properati") +  
  scale_fill_manual(values = c("goldenrod2", "indianred2")) +  
  facet_grid(operation_type~.)
```

### 3.5. RELACIÓN ENTRE VARIABLE NUMÉRICA Y CATEGÓRICA: GRÁFICO DE BARRAS85



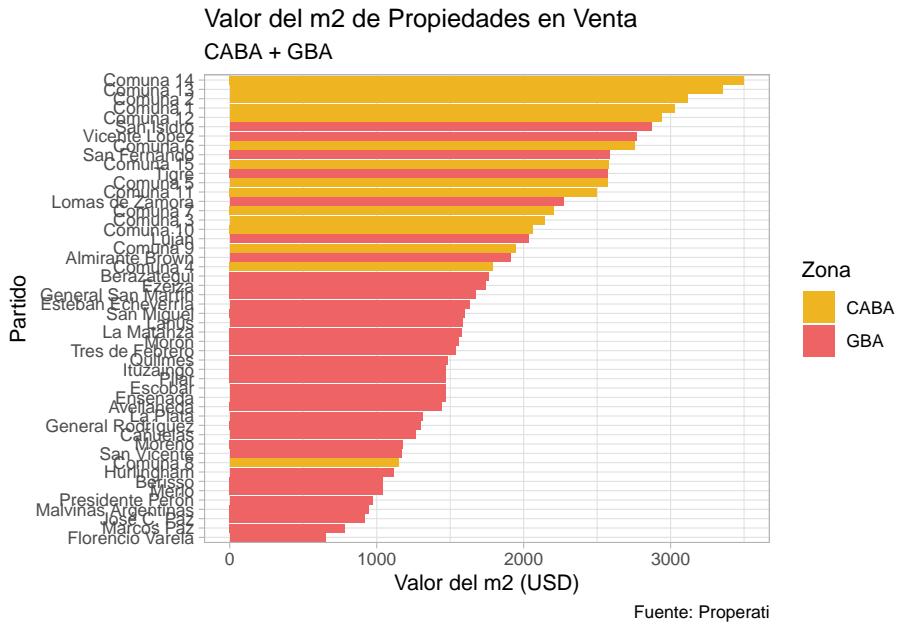
Vemos que:

- Los monoambientes publicados en CABA triplican los publicados en GBA.
- Para ambas operaciones, en CABA predominan los 2 y 3 ambientes.
- En las ventas de GBA predominan los 2, 3 y 4 ambientes.
- Para ambas zonas, las propiedades con más de 5 ambientes solo se encuentran en venta (no en alquiler).

Ahora entremos más en detalle y analicemos los datos por partido. Para eso filtremos por **Ventas** e incorporemos un valor weight: **precio del m<sup>2</sup>**. Como vimos que hay varios outliers en los datos del valor del m<sup>2</sup> que pueden afectar el promedio, utilizaremos la mediana:

```
ggplot(datos_amba %>%
  group_by(provincia, partido, operation_type) %>%
  summarise(cantidad=n(),
            price_m2=median(price/surface_covered)) %>%
  filter(operation_type=="Venta"))+
  geom_bar(aes(x=reorder(partido, price_m2), weight=price_m2, fill=provincia)) +
  labs(title="Valor del m2 de Propiedades en Venta",
       subtitle="CABA + GBA",
       fill="Zona",
       x="Partido",
       y="Valor del m2 (USD)",
       caption="Fuente: Properati") +
```

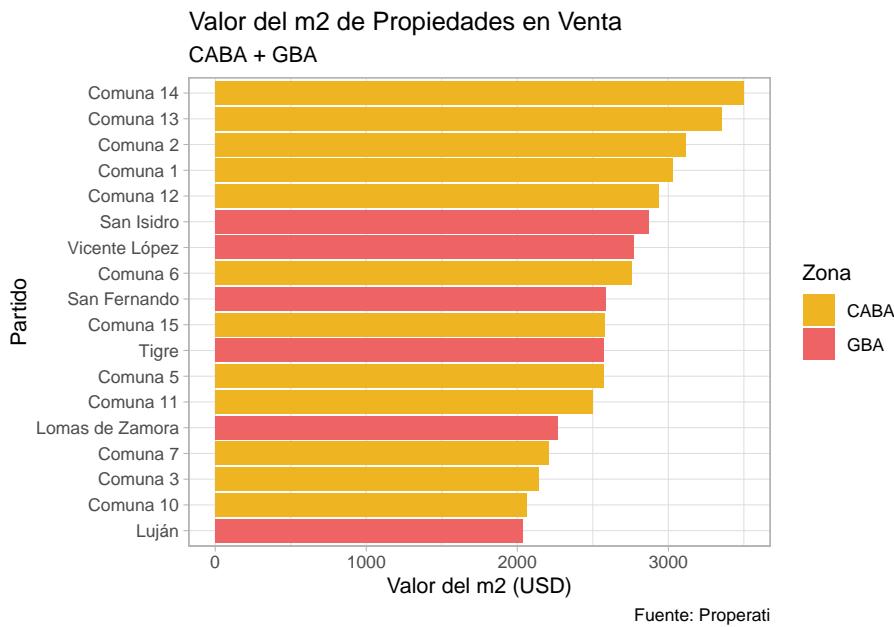
```
scale_fill_manual(values = c("goldenrod2", "indianred2")) +
theme_light()+
coord_flip()
```



Se ve un poco empastado y es difícil de leer. Probemos filtrando solo aquellos donde el valor del m<sup>2</sup> es mayor a 2.000 USD:

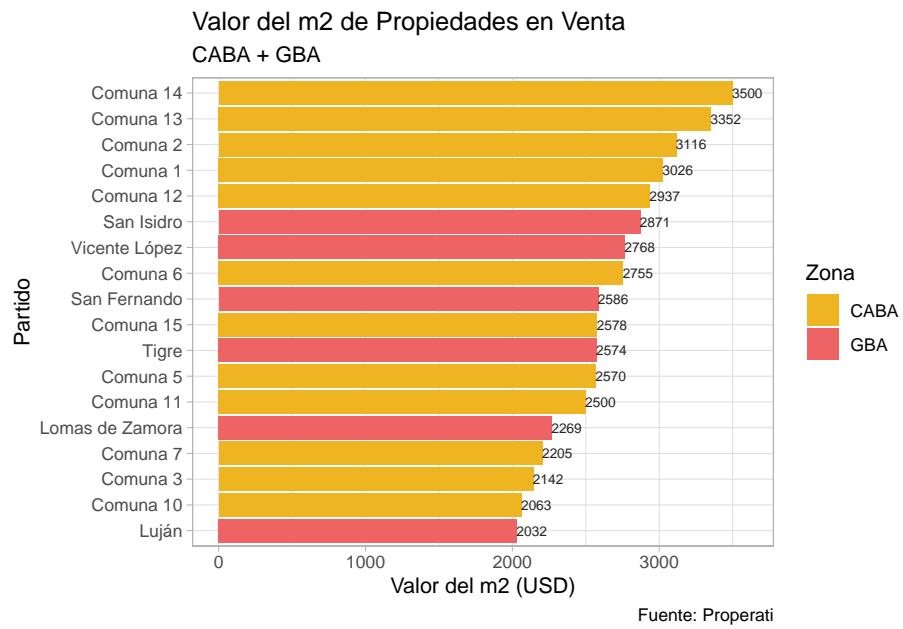
```
ggplot(datos_amba %>%
  group_by(provincia, partido, operation_type) %>%
  summarise(cantidad=n(), +
            price_m2=median(price/surface_covered)) %>%
  filter(operation_type=="Venta" & price_m2>=2000))+
  geom_bar(aes(x=reorder(partido, price_m2), weight=price_m2, fill=provincia)) +
  labs(title="Valor del m2 de Propiedades en Venta",
       subtitle="CABA + GBA",
       fill="Zona",
       x="Partido",
       y="Valor del m2 (USD)",
       caption="Fuente: Properati") +
  scale_fill_manual(values = c("goldenrod2", "indianred2")) +
  theme_light() +
  coord_flip()
```

### 3.5. RELACIÓN ENTRE VARIABLE NUMÉRICA Y CATEGÓRICA: GRÁFICO DE BARRAS<sup>87</sup>



Agreguemos etiquetas con `geom_text()` que nos faciliten la lectura:

```
ggplot(datos_amba %>%
  group_by(provincia, partido, operation_type) %>%
  summarise(cantidad=n(),
            price_m2=median(price/surface_covered)) %>%
  filter(operation_type=="Venta" & price_m2>=2000))+ 
  geom_bar(aes(x=reorder(partido, price_m2), weight=price_m2, fill=provincia)) +
  geom_text(aes(x=partido, y=price_m2+100, label=as.integer(price_m2)), size=2.5, color="gray14")+
  labs(title="Valor del m2 de Propiedades en Venta",
       subtitle="CABA + GBA",
       fill="Zona",
       x="Partido",
       y="Valor del m2 (USD)",
       caption="Fuente: Properati") +
  scale_fill_manual(values = c("goldenrod2", "indianred2")) +
  theme_light() +
  coord_flip()
```



## Capítulo 4

# INFORMACIÓN GEOGRÁFICA Y MAPAS

Ahora nos toca **visualizar en un mapa toda la información** que estuvimos analizando en los módulos anteriores. Para esto, vamos a trabajar con la información geográfica que contiene nuestro dataset y vamos agregar otras fuentes de datos geográficos.

Pero antes de empezar, ¿A qué nos referimos cuando hablamos de **Sistemas de Información Geográfica** o **SIG**? Bueno, nos referimos a las herramientas informáticas que nos permiten ubicar y analizar un conjunto de datos en lugares específicos del territorio (georreferenciar).

Para poder hacer uso de los SIG necesitamos contar con información geográfica en nuestro dataset, es decir, que además de la información que ya vimos que puede haber en un dataset tradicional, se sume un componente espacial en cada registro (partido, barrio, manzana, calle o directamente las coordenadas X e Y).

Las herramientas que nos permitirán visualizar toda esta información serán los **mapas**, que son nada más y nada menos que representaciones planas, reducidas y simplificadas de la tierra que nos dan la posibilidad cruzar y relacionar datos en el espacio. Es decir que, mantienen una relación ordenada en el traspaso de puntos ubicados en la superficie curva de la tierra a puntos ubicados en la superficie plana de los mapas. Esto es posible a partir del uso de sistemas de coordenadas proyectadas.

En R hay varios paquetes de funciones que nos permiten manipular este tipo de información, entre los que se encuentra **sf**, que lo aprenderemos hoy. Para comenzar a utilizarlo vamos a tener que instalarlo y luego activarlo con **library()** al igual que lo veníamos haciendo con **tidyverse**:

```
library(tidyverse)
#install.packages(sf)
library(sf)
```

Como verán, activamos los 2 paquetes porque ambos presentan funciones que son necesarias a la hora de mapear información.

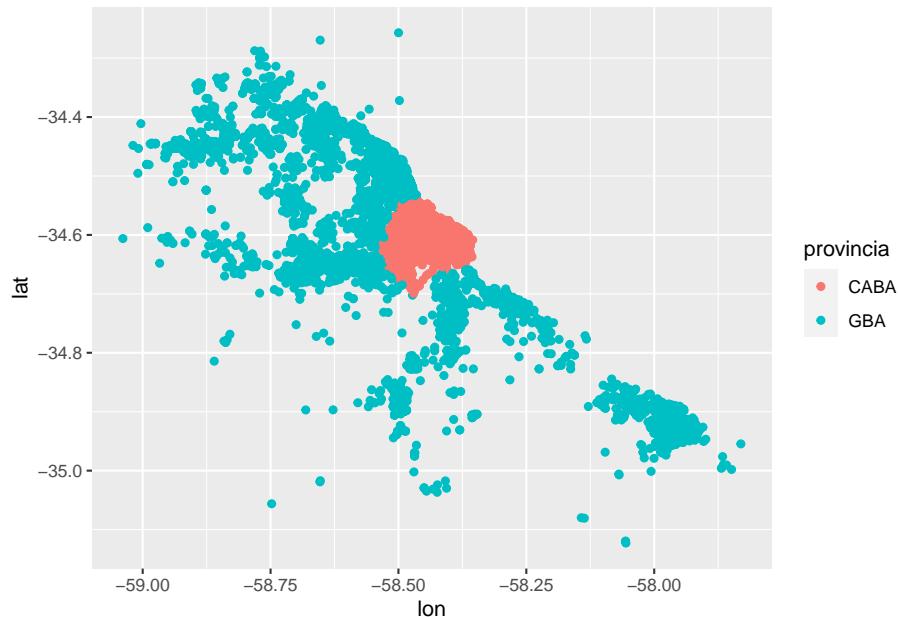
## 4.1 Analizar datos espaciales

Llegó el momento de conocer y analizar datos espaciales. Para esto, como hicimos en cada clase, volvamos a cargar nuestros datos de Properati:

```
datos_amba <- read.csv("data/amba_properati.csv")
```

Y repasemos el gráfico que generamos la clase pasada con `geom_point()` asignando la variable longitud (lon) en el eje X y la latitud (lat) en el Y:

```
ggplot(datos_amba) +
  geom_point(aes(x=lon, y=lat, color=provincia))
```



Con mucha imaginación uno puede darse cuenta que los datos tienen la forma de AMBA, pero está faltando información que nos facilite la lectura del mismo.

Para esto sumemos un dataset más a nuestro proyecto que en este caso, será un dataset espacial en formato shapefile (SHP) con las geometrías correspondientes a todos los partidos de AMBA. Pueden descargarlo (es un .zip que contiene el SHP) en el siguiente link: <https://data.world/angie-scetta/partidos-amba>

*Recomendación: Al igual que con el csv que trabajamos en las clases anteriores, al descargar el shape deberán moverlo de la carpeta “Descargas” a la carpeta llamada “data” dentro del Proyecto donde estan trabajando.*

Para poder cargar nuestros datos espaciales en formato shp utilizaremos la función `st_read()` de la siguiente forma:

```
partidos_amba <- st_read("data/partidos_amba.shp")
```

```
## Reading layer `partidos_amba' from data source `E:\03-OTROS\SCA-CURSOS-2020\sca-big-data-urban
## Simple feature collection with 48 features and 3 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:            xmin: -59.3392 ymin: -35.23893 xmax: -57.70946 ymax: -34.23007
## CRS:             4326
```

Veamos que información contiene:

```
head(partidos_amba)
```

```
## Simple feature collection with 6 features and 3 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:            xmin: -59.05579 ymin: -34.91331 xmax: -58.27953 ymax: -34.26732
## CRS:             4326
##   nombre provincia area_km2           geometry
## 1 Avellaneda     GBA    57.25 MULTIPOLYGON (((-58.33444 -...
```

#	nombre	provincia	area_km2	geometry
1	Avellaneda	GBA	57.25	MULTIPOLYGON (((-58.33444 -...
2	Tigre	GBA	381.99	MULTIPOLYGON (((-58.5167 -3...
3	Pilar	GBA	382.95	MULTIPOLYGON (((-58.90312 -...
4	Moreno	GBA	186.36	MULTIPOLYGON (((-58.82401 -...
5	Merlo	GBA	173.97	MULTIPOLYGON (((-58.72917 -...
6	La Matanza	GBA	328.26	MULTIPOLYGON (((-58.52885 -...

Las primeras 3 columnas presentan el nombre del partido, la provincia y el área en km<sup>2</sup>. Hasta acá son datos muy similares a las que ya veníamos encontrando en los dataset tradicionales; sin embargo, aparece una 4ta columna llamada “geometry” que hasta ahora no la habíamos visto y es donde se aloja la geometría de cada uno de los registros. La información de este campo es la que hace que el dataset sea espacial.

Si queda alguna duda, podemos utilizar la función `class()` para ver con qué tipo de datos estamos trabajando:

```
class(datos_amba)
```

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

Tal como lo imaginábamos, datos\_amba es un simple dataframe.

```
class(partidos_amba)
```

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

Pero partidos\_amba, es un dataset espacial u objeto del tipo “sf”, que hace referencia a “simple features” por estar compuesto de geometrías bidimensionales (polígono, punto, línea, multipunto, multilínea, etc.).

Para poder visualizar toda esta información plasmada en un mapa vamos a utilizar nuevamente `ggplot()` pero como en esta oportunidad queremos sumar capas geográficas (`sf`) trabajaremos con `geom_sf()`. Veamos un ejemplo:

```
ggplot(partidos_amba)+  
  geom_sf()
```

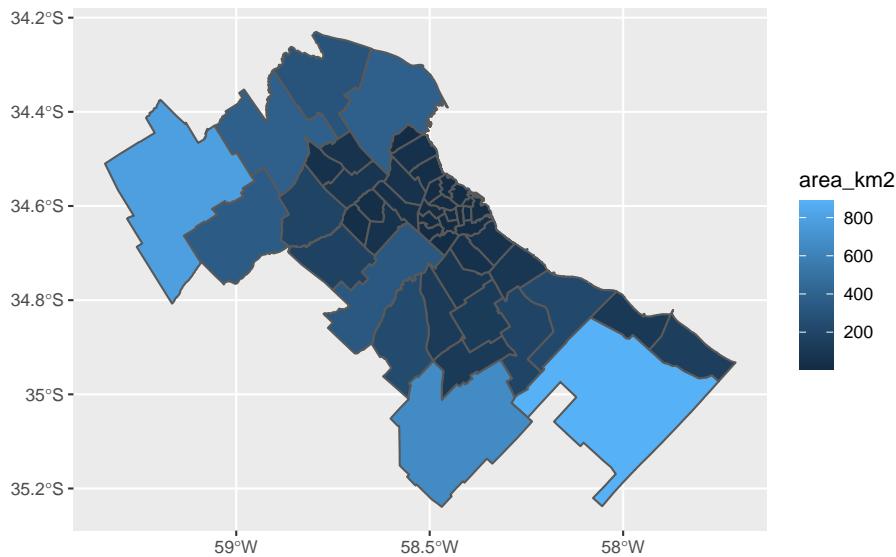


Como habrán notado, la lógica en la estructura del chunk que utilizamos para hacer este mapa es la misma del capítulo anterior, donde dentro del `ggplot()`

asignamos el dataset y luego sumamos la capa a graficar/mapear (en este caso un objeto sf).

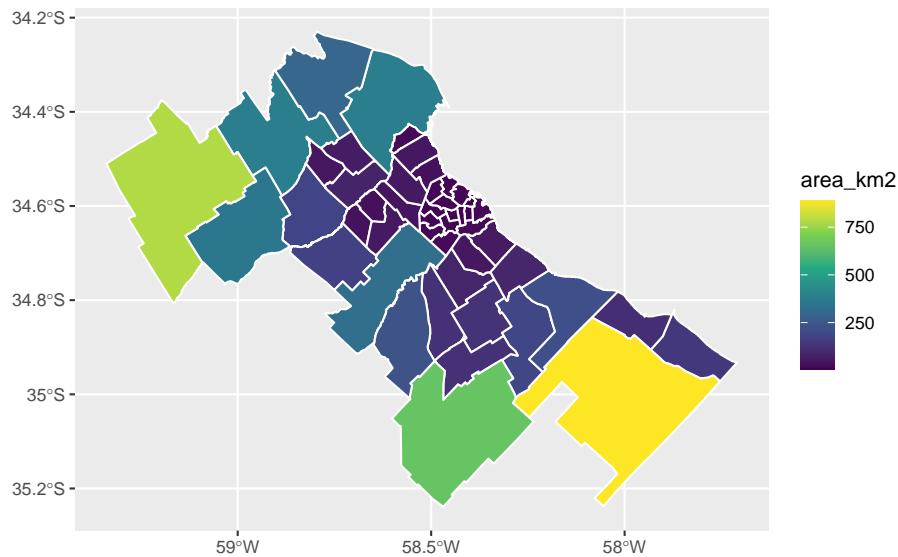
En el mapa podemos ver como las geometrías de cada registro del dataset espacial son polígonos que representan los partidos de AMBA. Probemos agregar algún atributo estético `aes()`:

```
ggplot(partidos_amba)+  
  geom_sf(aes(fill=area_km2))
```



Ya tenemos nuestro primer **mapa coroplético**. Probemos ajustando algunas cuestiones estéticas: cambiemos la paleta (y su escala) y el color del borde de los polígonos.

```
ggplot(partidos_amba)+  
  geom_sf(aes(fill=area_km2), color="white") +  
  scale_fill_viridis_c(breaks=c(0,250,500,750,1000))
```

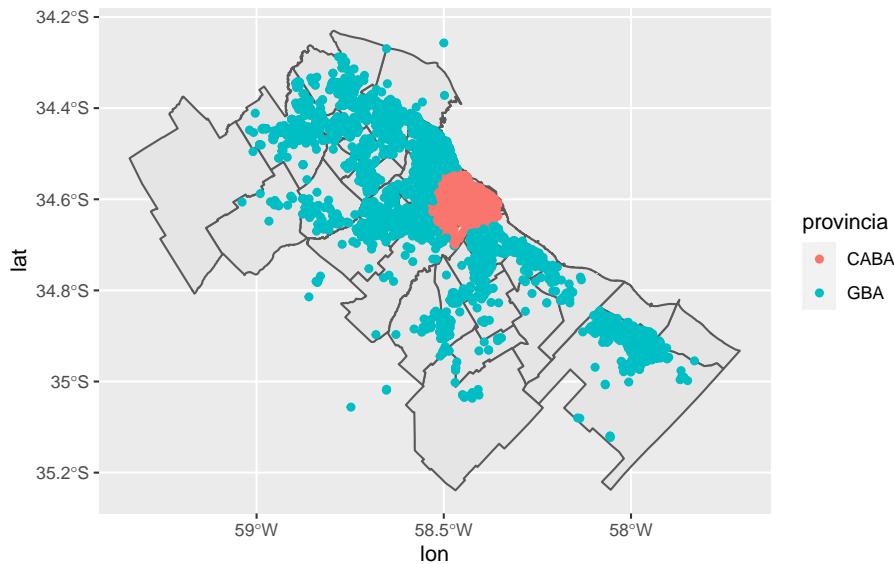


Y ahora si queremos visualizar en un mismo mapa el dataset espacial (`partidos_amba`) y el dataset tradicional (`datos_amba`) tenemos que tener en cuenta 2 cosas:

1. Cuando utilizamos 2 capas (en este caso `geom_sf()` y `geom_point()`), tenemos que asignarle a cada una el dataset dentro de la capa y `ggplot()` queda “vacío”.
2. Las capas se grafican en el mismo orden que se escribe el código, es decir que la que agregamos primero (en este caso `partidos_amba`) será el fondo de la que agreguemos luego (`datos_amba`).

Veamos esto en detalle:

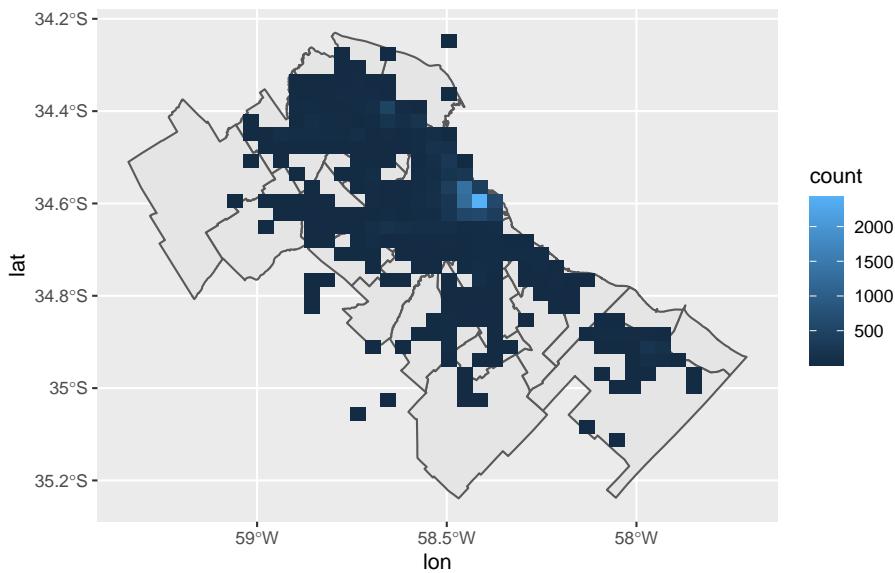
```
ggplot()+
  geom_sf(data=partidos_amba)+
  geom_point(data=datos_amba, aes(x=lon, y=lat, color=provincia))
```



En el mapa anterior hay tantos puntos que nos resulta muy difícil poder encontrar patrones que nos permitan entender donde se concentran más cantidad de puntos y donde menos. En estos casos podemos recurrir a los **mapas de densidad de puntos** que nos ayudarán a encontrar los “hot spots”.

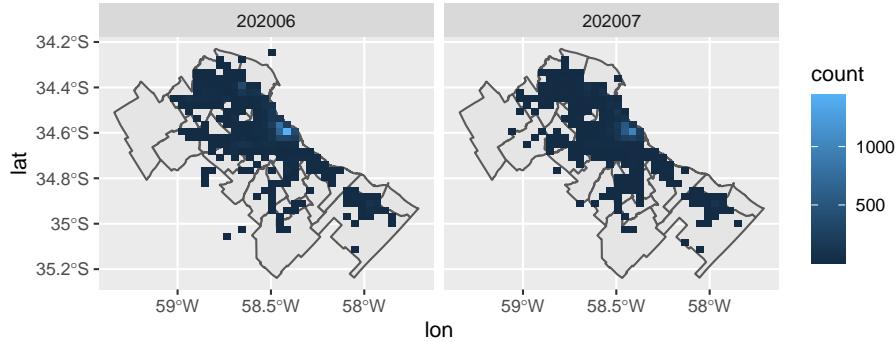
En R hay varias formas de mapear esto, pero hoy optaremos por `geom_bin2d()`, que divide el plano en una grilla y cuenta la cantidad de puntos que aparecen en cada celda.

```
ggplot()+
  geom_sf(data=partidos_amba)+
  geom_bin2d(data = datos_amba, aes(x = lon, y = lat))
```



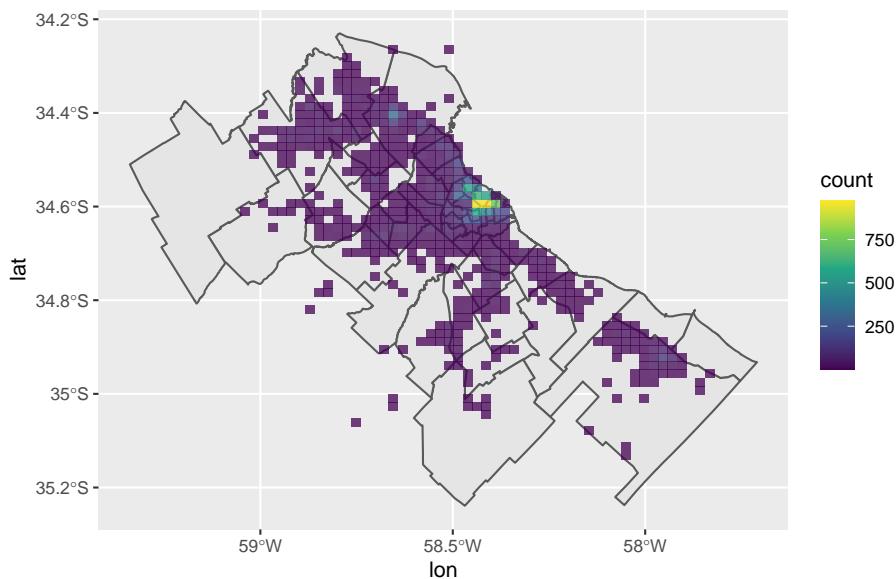
Ahora si nos encontramos con un hot spot que antes no veíamos. Se ve muy claro como en Palermo se concentran la mayor cantidad de propiedades publicadas en Junio y Julio en Properati. Pero, ¿Existirá una diferencia entre los 2 meses? Respondamos esto con un facetado por created\_on:

```
ggplot()+
  geom_sf(data=partidos_amba)+
  geom_bin2d(data = datos_amba, aes(x = lon, y = lat))+
  facet_grid(~created_on)
```



Claramente los patrones de ambos meses son muy parecidos, así que mantengamos un único mapa pero cambiemos el color y el tamaño de los bins (celdas):

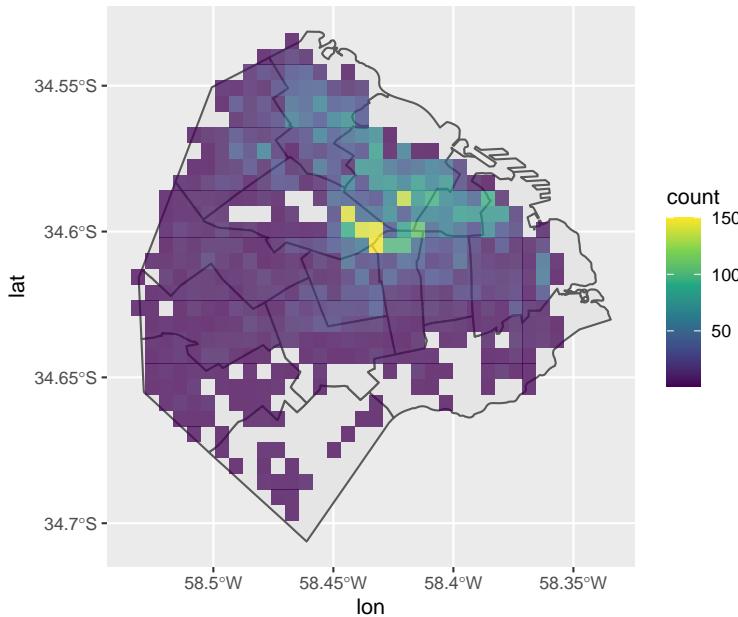
```
ggplot()+
  geom_sf(data=partidos_amba)+
  geom_bin2d(data = datos_amba, aes(x = lon, y = lat), alpha=0.75, bins=50)+
  scale_fill_viridis_c()
```



Y cómo hacemos si queremos ver este mapa con un “zoom” solo de CABA?

Tenemos que filtrar ambas capas así:

```
ggplot()+
  geom_sf(data=filter(partidos_amba, provincia=="CABA"))+
  geom_bin2d(data = filter(datos_amba, provincia=="CABA"), aes(x = lon, y = lat), alpha=0.8)
```



Viendo el mapa anterior nos cambia un poco la percepción y ya no parecería ser solo la Comuna 14 la que tiene la mayor densidad de propiedades, sino que se ve una zona con gran densidad en la Comuna 15 (por Villa Crespo).

Pero bueno, a pesar de haber llegado a un mapa que nos permitió sacar muchas conclusiones, aún estamos trabajando con 2 dataset separados que no se han unido. ¡Veamos como cruzar y unificar todos estos datos!

## 4.2 Cruzar datos tradicionales y espaciales

Es muy común que a la hora de trabajar con datos no encontramos toda la información que necesitamos en un solo dataset. Frente a esto, lo que se hace es unir/cruzar datos provenientes de diferentes fuentes de información. En este apartado veremos como realizar estos cruces entre datos tradicionales y espaciales.

Para poder unir 2 set de datos ambos tienen que tener algo en común (alguna columna como por ejemplo un ID), sino sería imposible que R entienda que tiene que unir con qué. Por lo tanto, antes de unirlos deberemos manipularlos y realizarles diferentes tipos de transformaciones que nos permitan llegar a generar esta variable en común.

Por ejemplo en nuestro caso, si queremos unir algún dato a nuestro dataset espacial de partidos, deberíamos tener un valor único por cada nombre de partido (que funciona como el ID de mi dataset espacial).

Trabajemos con las propiedades en venta y manipulemos un poco el dataset tradicional para que podamos unirlos:

```
datos_amba_venta <- datos_amba %>%
  filter(operation_type=="Venta") %>%
  group_by(partido) %>%
  summarise(cantidad=n(),
            valor_m2=mean(price/surface_covered))

head(datos_amba_venta)

## # A tibble: 6 x 3
##   partido      cantidad  valor_m2
##   <fct>        <int>    <dbl>
## 1 Almirante Brown     24    1824.
## 2 Avellaneda         67    1487.
## 3 Berazategui        47    1787.
## 4 Berisso             3     952.
## 5 Cañuelas            3    1293.
## 6 Comuna 1            542   3526.
```

Bien, ahora que tenemos un dataset de 50 observaciones/registros (partidos) donde para cada partido hay 2 columnas con valores asociados (cantidad y valor del m2), ya estamos en condiciones de hacer una unión con el dataset espacial.

Para esto utilizaremos la función `left_join()`:

```
partidos_amba <- left_join(partidos_amba, datos_amba_venta, by=c("nombre"="partido"))

head(partidos_amba)

## Simple feature collection with 6 features and 5 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:            xmin: -59.05579 ymin: -34.91331 xmax: -58.27953 ymax: -34.26732
## CRS:             4326
##           nombre provincia area_km2 cantidad valor_m2
## 1     Avellaneda      GBA     57.25     67 1486.522
## 2       Tigre          GBA    381.99    887 2674.501
## 3       Pilar          GBA    382.95    493 1680.706
## 4      Moreno          GBA    186.36     64 1202.036
## 5      Merlo           GBA    173.97     83 1146.151
## 6 La Matanza          GBA    328.26    110 1592.995
##           geometry
```

```
## 1 MULTIPOLYGON (((-58.33444 -...
```

```
## 2 MULTIPOLYGON (((-58.5167 -3...
```

```
## 3 MULTIPOLYGON (((-58.90312 -...
```

```
## 4 MULTIPOLYGON (((-58.82401 -...
```

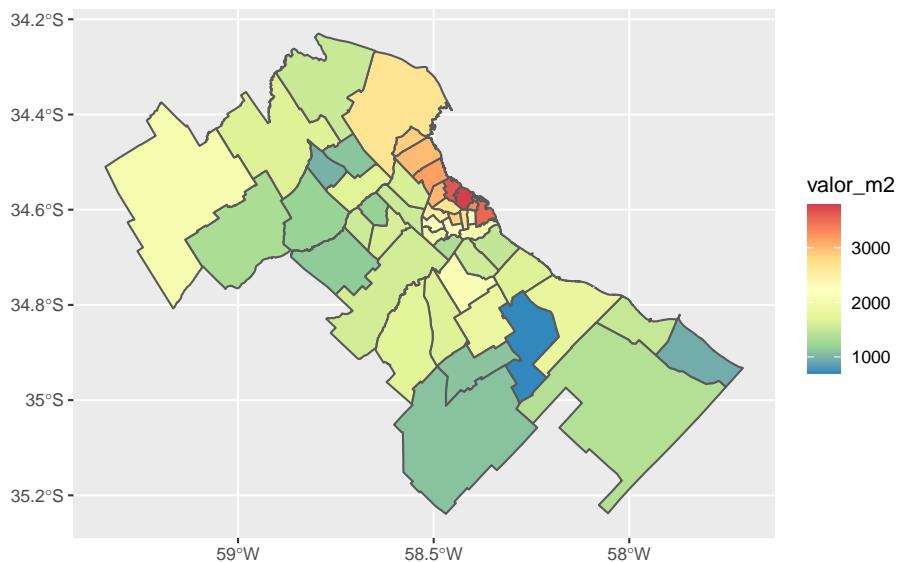
```
## 5 MULTIPOLYGON (((-58.72917 -...
```

```
## 6 MULTIPOLYGON (((-58.52885 -...
```

Tal como esperábamos, se agregaron 2 nuevas columnas: cantidad y valor\_m2.

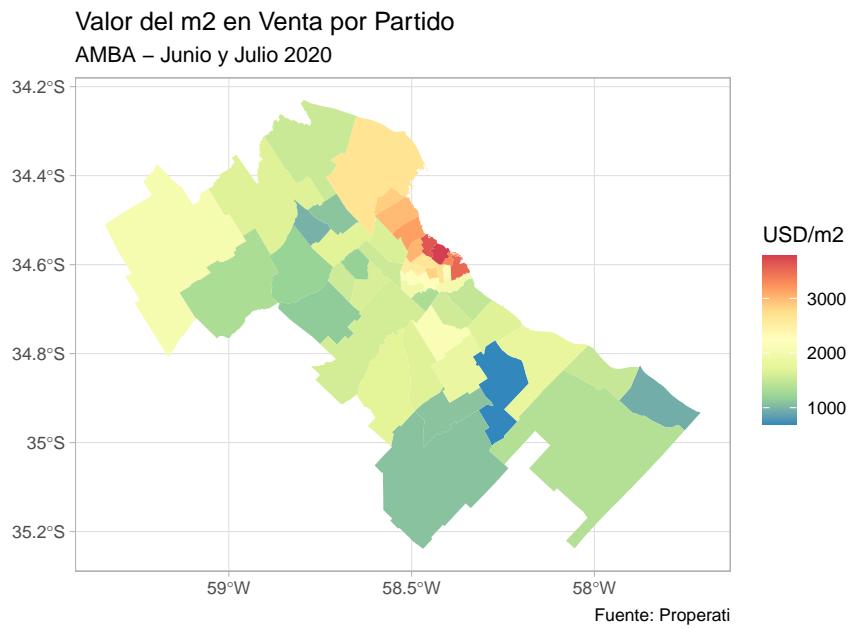
Veamos esto en un mapa:

```
ggplot(partidos_amba)+  
  geom_sf(aes(fill=valor_m2))+  
  scale_fill_distiller(palette = "Spectral")
```



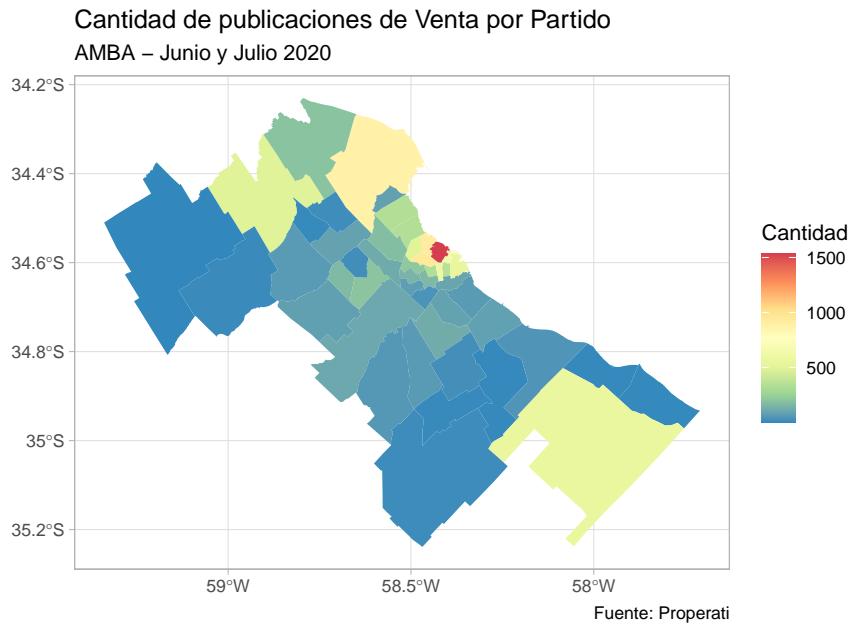
Al igual que en los gráficos, en los mapas pueden agregarse etiquetas (título, subtítulo, etc) y cambiar el aspecto (theme):

```
ggplot() +  
  geom_sf(data=partidos_amba, aes(fill=valor_m2), color=NA) +  
  labs(title = "Valor del m2 en Venta por Partido",  
       subtitle = "AMBA - Junio y Julio 2020",  
       fill = "USD/m2",  
       caption= "Fuente: Properati") +  
  scale_fill_distiller(palette = "Spectral") +  
  theme_light()
```



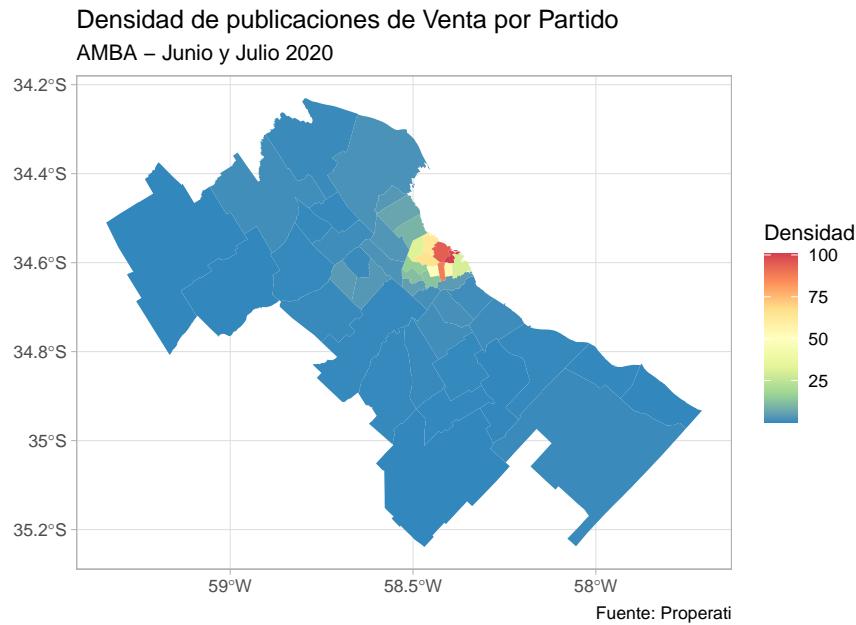
Ahora mapiemos la otra nueva variable: cantidad de propiedades publicadas para venta.

```
ggplot()+
  geom_sf(data=partidos_amba, aes(fill=cantidad), color=NA) +
  labs(title = "Cantidad de publicaciones de Venta por Partido",
       subtitle = "AMBA - Junio y Julio 2020",
       fill = "Cantidad",
       caption= "Fuente: Properati") +
  scale_fill_distiller(palette = "Spectral") +
  theme_light()
```



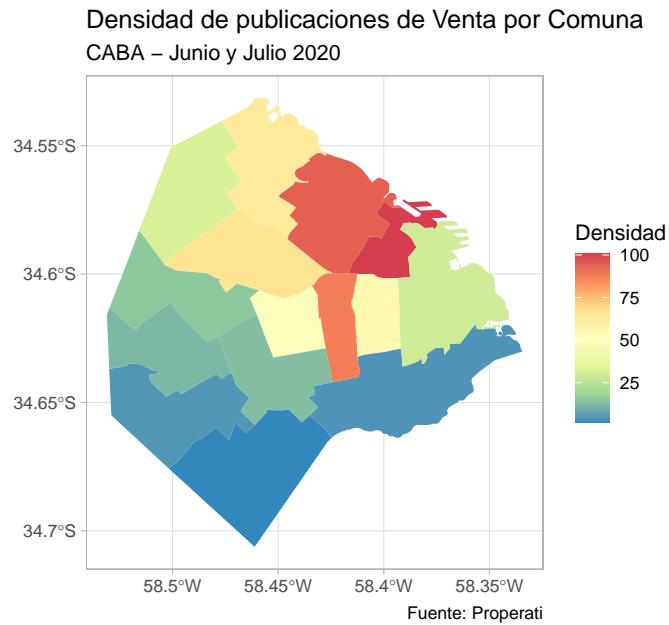
En el mapa se ve como la Comuna 14 tiene el valor más alto, sin embargo, para que estos datos sean comparables entre todos los partidos, es necesario que calculemos una densidad relacionando la cantidad de propiedades y la superficie ( $\text{km}^2$ ) de cada uno:

```
ggplot()+
  geom_sf(data=partidos_amba, aes(fill=cantidad/area_km2), color=NA) +
  labs(title = "Densidad de publicaciones de Venta por Partido",
       subtitle = "AMBA – Junio y Julio 2020",
       fill = "Densidad",
       caption= "Fuente: Properati") +
  scale_fill_distiller(palette = "Spectral") +
  theme_light()
```



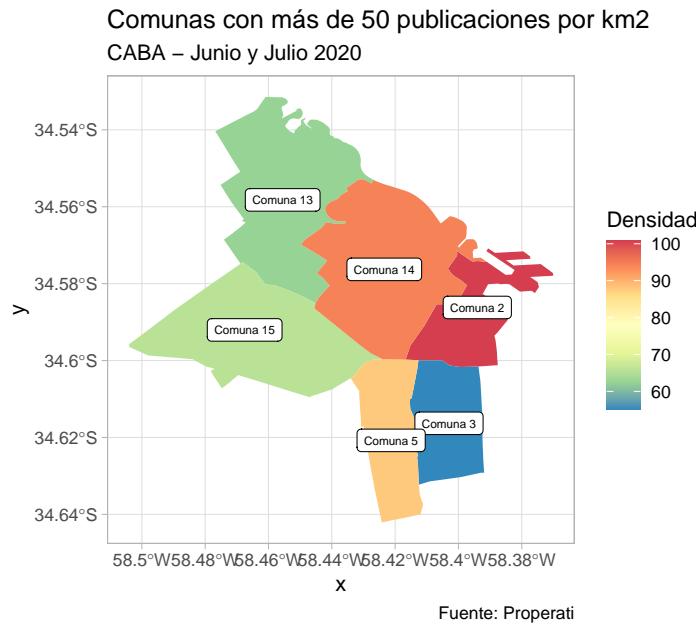
El mapa cambia bastante, ahora todos los partidos de AMBA son color azul, es decir que tienen baja densidad y los de CABA se ven coloreados. Hagamos un zoom en CABA y veamos esto en detalle:

```
ggplot()+
  geom_sf(data=filter(partidos_amba, provincia=="CABA"), aes(fill=cantidad/area_km2),
          labs(title = "Densidad de publicaciones de Venta por Comuna",
               subtitle = "CABA - Junio y Julio 2020",
               fill = "Densidad",
               caption= "Fuente: Properati") +
  scale_fill_distiller(palette = "Spectral") +
  theme_light()
```



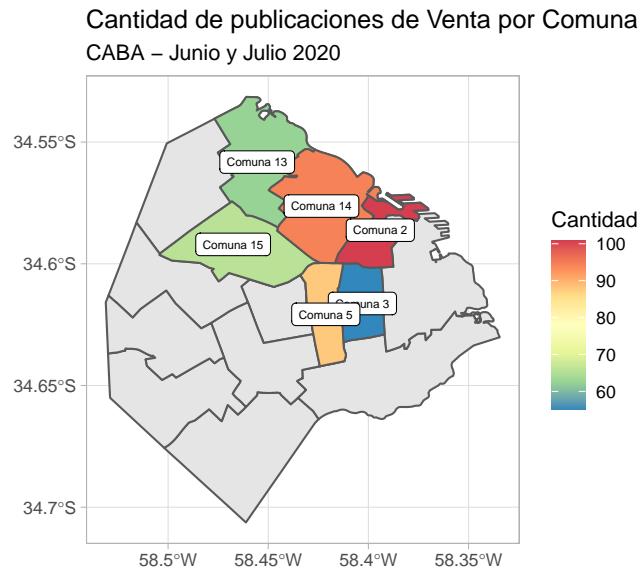
Podemos ver que aparece la Comuna 5 como la más densa seguida por la 14 y la 5. Sin embargo hay varias que tienen una densidad mayor a 50 publicaciones por km<sup>2</sup>. Veamos cuáles son:

```
ggplot()+
  geom_sf(data=filter(partidos_amba, provincia=="CABA" & cantidad/area_km2>=50), aes(fill=cantidad))
  geom_sf_label(data=filter(partidos_amba, provincia=="CABA" & cantidad/area_km2>=50), aes(label=cantidad))
  labs(title = "Comunas con más de 50 publicaciones por km2",
       subtitle = "CABA - Junio y Julio 2020",
       fill = "Densidad",
       caption= "Fuente: Properati") +
  scale_fill_distiller(palette = "Spectral") +
  theme_light()
```



Y por último agreguémosle el mapa de fondo así las comunas no quedan flotando:

```
ggplot()+
  geom_sf(data=filter(partidos_amba, provincia=="CABA")) +
  geom_sf(data=filter(partidos_amba, provincia=="CABA" & cantidad/area_km2>=50), aes(fill= cantidad/area_km2>=50),
  geom_sf_label(data=filter(partidos_amba, provincia=="CABA" & cantidad/area_km2>=50),
  labs(title = "Cantidad de publicaciones de Venta por Comuna",
       subtitle = "CABA - Junio y Julio 2020",
       fill = "Cantidad",
       x="",
       y="",
       caption= "Fuente: Properati") +
  scale_fill_distiller(palette = "Spectral") +
  theme_light()
```



Fuente: Properati

### Variables Categóricas en el Mapa

Hasta acá todos los mapas que desarrollamos tomaron color a partir de una variable numérica (cantidad o valor del m<sup>2</sup>), pero ¿Qué pasa si lo que queremos mapear es una variable categórica, cómo por ejemplo la tipología que más aparece publicada por partido?

Para resolver esta incógnita debemos generar una nueva variable que contenga la información y pueda unirse a nuestro dataset espacial `partidos_amba`:

```
datos_amba_tipologia <- datos_amba %>%
  group_by(partido, property_type) %>%
  summarise(cant_max=n()) %>%
  filter(cant_max==max(cant_max))

head(datos_amba_tipologia)

## # A tibble: 6 x 3
## # Groups:   partido [6]
##   partido      property_type cant_max
##   <fct>        <fct>          <int>
## 1 Almirante Brown Casa            19
## 2 Avellaneda    Departamento   47
## 3 Berazategui  Departamento   35
## 4 Berisso       Casa             3
```

```
## 5 Cañuelas      Casa      3
## 6 Comuna 1     Departamento 739
```

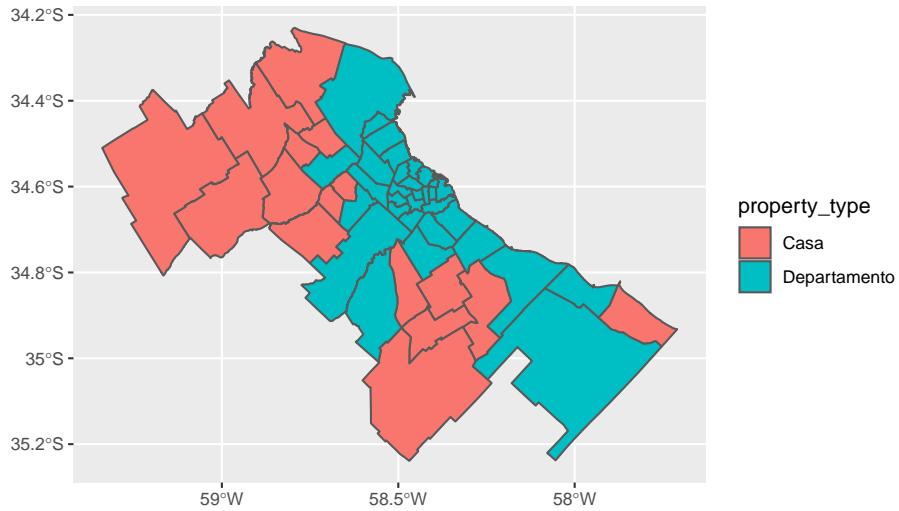
```
partidos_amba <- partidos_amba %>%
  left_join(datos_amba_tipologia, by=c("nombre"="partido"))
```

```
head(partidos_amba)
```

```
## Simple feature collection with 6 features and 7 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:            xmin: -59.05579 ymin: -34.91331 xmax: -58.27953 ymax: -34.26732
## CRS:             4326
##           nombre provincia area_km2 cantidad valor_m2 property_type cant_max
## 1 Avellaneda      GBA    57.25      67 1486.522 Departamento      47
## 2 Tigre          GBA   381.99     887 2674.501 Departamento     898
## 3 Pilar          GBA   382.95     493 1680.706 Casa      353
## 4 Moreno         GBA   186.36      64 1202.036 Casa      50
## 5 Merlo          GBA   173.97      83 1146.151 Casa      54
## 6 La Matanza     GBA   328.26     110 1592.995 Departamento     73
##           geometry
## 1 MULTIPOLYGON (((-58.33444 -...
```

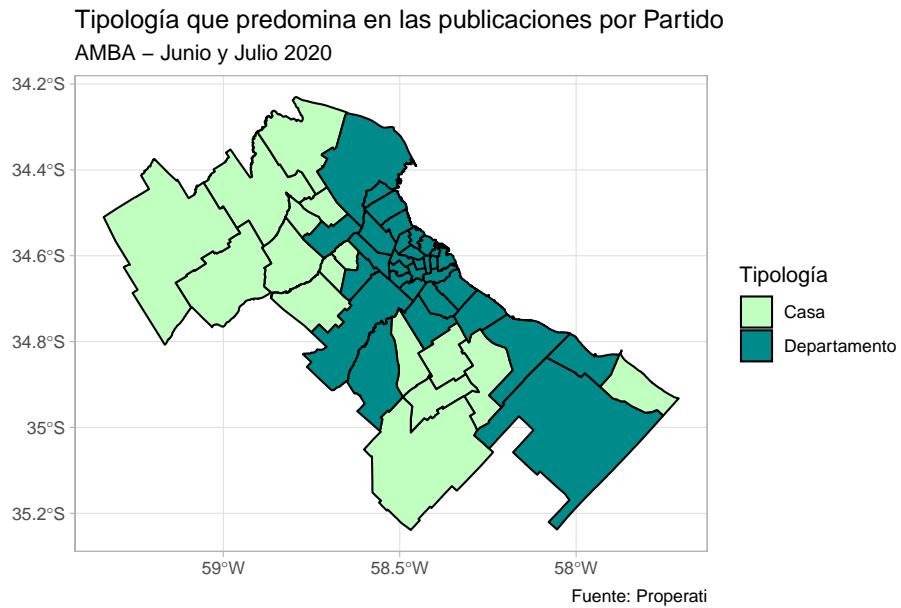
Efectivamente, se sumaron 2 nuevas columnas: property\_type y cant\_max. Ahora si, mapiemos por la tipología que más aparece:

```
ggplot(partidos_amba) +
  geom_sf(aes(fill=property_type))
```



Y modifiquemos su estética como ya aprendimos:

```
ggplot(partidos_amba)+  
  geom_sf(aes(fill=property_type), color="black") +  
  labs(title = "Tipología que predomina en las publicaciones por Partido",  
       subtitle = "AMBA - Junio y Julio 2020",  
       fill = "Tipología",  
       caption= "Fuente: Properati") +  
  scale_fill_manual(values=c("darkseagreen1", "cyan4"))+  
  theme_light()
```



### 4.3 Cruzar datos espaciales

Muchas veces nos vamos a encontrar con que queremos unir 2 dataset pero no tienen columnas en común y es imposible generarlas; pero hay **algo que si tienen en común y que nos permitirá unirlos: la ubicación en el espacio**. A esto nos referimos cuando hablamos de “Spatial Join” o “Unión Espacial” y en R lo haremos a partir de la función `st_join()` que forma parte del paquete `sf`.

Pero antes de seguir, nos está faltando un segundo dataset espacial ya que hasta ahora solo tenemos uno. Como primer paso vamos a tener que transformar nuestro dataset tradicional “datos\_amba” a un dataset espacial utilizando la función `st_as_sf()` de la siguiente forma:

```
datos_amba_geo <- datos_amba %>%
  st_as_sf(coords = c("lon", "lat"), crs = 4326)

head(datos_amba_geo)

## Simple feature collection with 6 features and 11 fields
## geometry type: POINT
## dimension: XY
## bbox: xmin: -58.52914 ymin: -34.66253 xmax: -58.3609 ymax: -34.53344
```

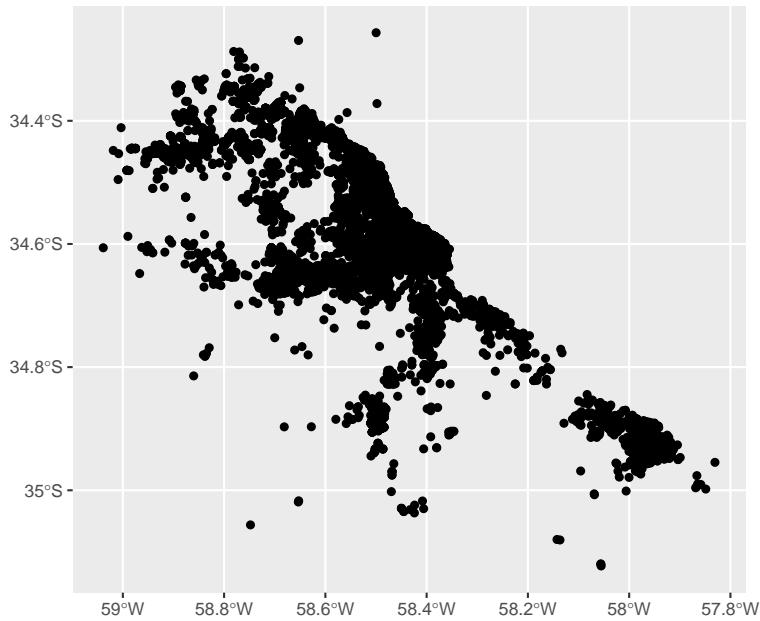
```

## CRS:           EPSG:4326
##   created_on provincia      partido rooms surface_total surface_covered price
## 1    202006      CABA     Comuna 7     1        40            37 22500
## 2    202006      CABA     Comuna 13    1        30            30 18000
## 3    202006      CABA     Comuna 13    1        31            29 17900
## 4    202006      CABA     Comuna 1     1        35            35 42000
## 5    202006      GBA  Vicente López    1        36            27 19000
## 6    202006      GBA  La Matanza     2        24            24 12000
##   currency
## 1      ARS
## 2      ARS
## 3      ARS
## 4      ARS
## 5      ARS
## 6      ARS
##                                title
## 1                          Departamento - Flores
## 2          Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3                          Departamento - Belgrano
## 4          Monoambiente con cochera. Zencyt. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6                           PH - Lomas Del Mirador
##   property_type operation_type      geometry
## 1 Departamento       Alquiler POINT (-58.46222 -34.61917)
## 2 Departamento       Alquiler POINT (-58.46652 -34.5546)
## 3 Departamento       Alquiler POINT (-58.46461 -34.56318)
## 4 Departamento       Alquiler POINT (-58.3609 -34.61836)
## 5 Departamento       Alquiler POINT (-58.49345 -34.53344)
## 6          PH       Alquiler POINT (-58.52914 -34.66253)

```

Ahora si, nuestro dataset datos\_amba se transformó en espacial y eso lo vemos con el reemplazo de sus columnas X e Y por una única columna llamada “geometry”. Hagamos un mapa y veamos que pasa:

```
ggplot(datos_amba_geo)+  
  geom_sf()
```



A simple vista son iguales que cuando hicimos el `geom_point()` pero la diferencia es que ahora son espaciales y ya estamos en condiciones de unirlos con otro dataset espacial. En este caso, como queremos generar información nueva, vamos a unirlos con un SHP que contenga información nueva que nuestros datos no tienen como por ejemplo los barrios de CABA a los que corresponde cada publicación.

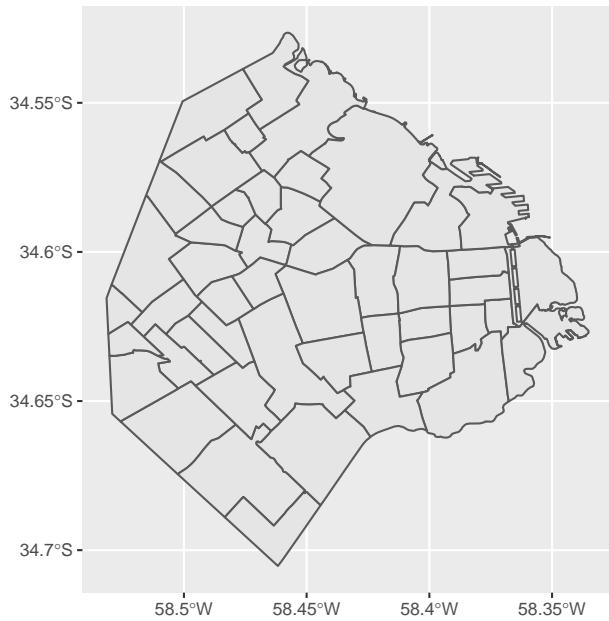
Para esto vamos a descargar el SHP de <https://data.world/angie-scetta/barrios-caba>, guardarlo en la carpeta “data” y luego cargarlo de la siguiente forma:

```
barrios_caba <- st_read("data/barrios_caba.shp")
```

```
## Reading layer `barrios_caba' from data source `E:\03-OTROS\SCA-CURSOS-2020\sca-big-data\barrios_caba.shp'
## Simple feature collection with 48 features and 1 field
## geometry type:  POLYGON
## dimension:      XY
## bbox:            xmin: -58.53152 ymin: -34.70529 xmax: -58.33515 ymax: -34.52649
## CRS:             4326
```

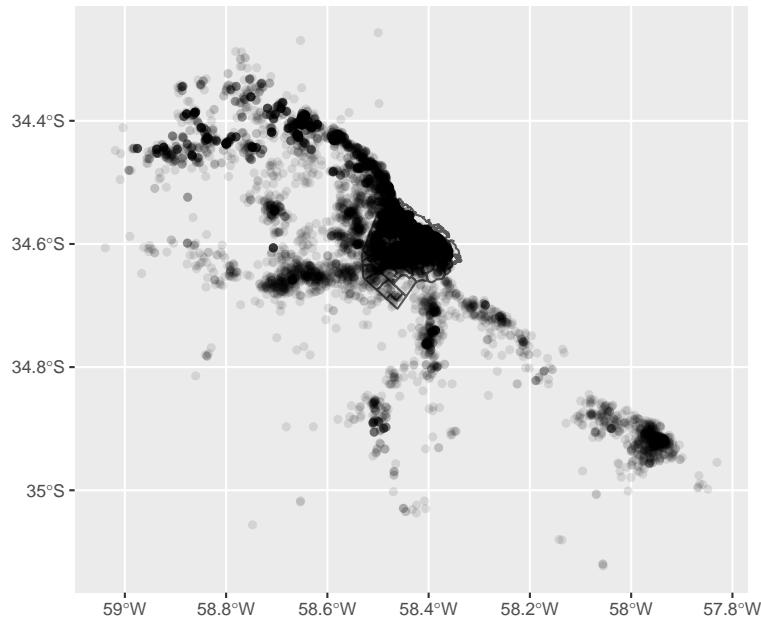
Veamos los polígonos:

```
ggplot(barrios_caba)+  
  geom_sf()
```



Efectivamente tienen forma de los 48 barrios de CABA. Superpongamos los datos de Properati que convertimos en el paso anterior para ver cuanto se “solapan” con el SHP de barrios:

```
ggplot()+
  geom_sf(data=barrios_caba)+
  geom_sf(data=datos_amba_geo, alpha=0.1)
```



Como mi SHP solo contiene los barrios de CABA, obviamente solo se solapa con los puntos que corresponden a propiedades ubicadas en CABA.

Ahora si, hagamos nuestra unión espacial con `st_join()` para que, cada registro de `datos_amba_geo` sume una nueva columna que indique a qué barrio pertenece:

```
datos_caba_geo <- st_join(datos_amba_geo, barrios_caba)
```

```
head(datos_caba_geo)
```

```
## Simple feature collection with 6 features and 12 fields
## geometry type:  POINT
## dimension:      XY
## bbox:            xmin: -58.52914 ymin: -34.66253 xmax: -58.3609 ymax: -34.53344
## CRS:             EPSG:4326
##   created_on provincia      partido rooms surface_total surface_covered price
## 1    202006     CABA     Comuna 7     1        40          37  22500
## 2    202006     CABA     Comuna 13    1        30          30  18000
## 3    202006     CABA     Comuna 13    1        31          29  17900
## 4    202006     CABA     Comuna 1     1        35          35  42000
## 5    202006     GBA Vicente López  1        36          27  19000
## 6    202006     GBA  La Matanza    2        24          24  12000
##   currency
## 1     ARS
```

```

## 2      ARS
## 3      ARS
## 4      ARS
## 5      ARS
## 6      ARS
##
##                                             title
## 1                                         Departamento - Flores
## 2             Retasado! Monoambiente en Nuñez, excelente ubicación!
## 3                                         Departamento - Belgrano
## 4             Monoambiente con cochera. Zencyty. Puerto Madero
## 5 Alquiler TORRE dpto de 1o2 ambientes - excelente luz y vista cochera optativa
## 6                                         PH - Lomas Del Mirador
##   property_type operation_type      BARRIO           geometry
## 1 Departamento     Alquiler      FLORES POINT (-58.46222 -34.61917)
## 2 Departamento     Alquiler      NUÑEZ POINT (-58.46652 -34.5546)
## 3 Departamento     Alquiler      BELGRANO POINT (-58.46461 -34.56318)
## 4 Departamento     Alquiler     PUERTO MADERO POINT (-58.3609 -34.61836)
## 5 Departamento     Alquiler      <NA> POINT (-58.49345 -34.53344)
## 6          PH        Alquiler      <NA> POINT (-58.52914 -34.66253)

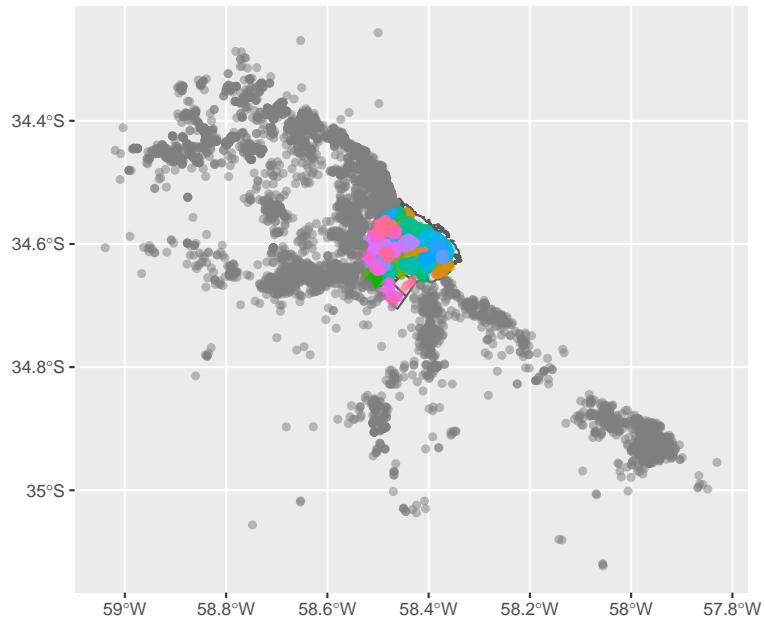
```

Como verán, a cada registro se le unió una columna con el nombre del barrio con el que solapan, y a aquellas propiedades ubicadas fuera de CABA se les asignó un valor nulo o NA. Veamos esto en un mapa:

```

ggplot()+
  geom_sf(data=barrios_caba)+
  geom_sf(data=datos_caba_geo, aes(color=BARRIO), alpha=0.5, show.legend = FALSE)

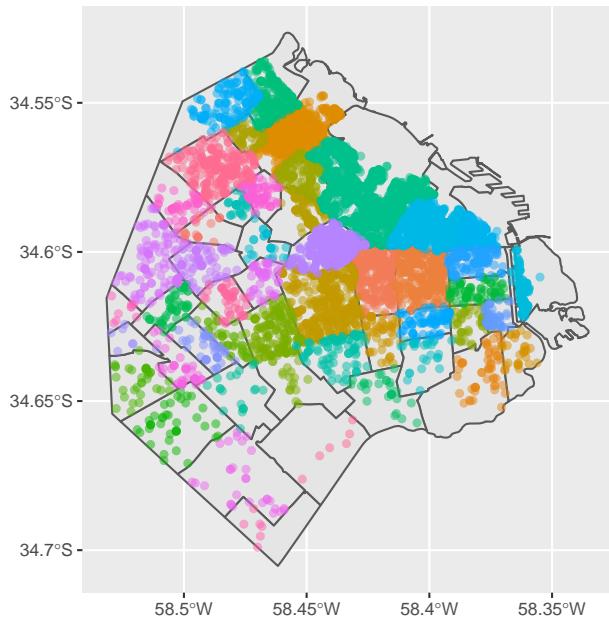
```



Se ve como todos los puntos que se ubican en CABA tienen color según sus barrios, pero los puntos fuera de CABA quedaron grises porque su valor es NA. Para “limpiar” esto filtremos todos los registros que tienen barrio asignado:

```
datos_caba_geo <- datos_caba_geo %>%
  filter(!is.na(BARRIO))
```

```
ggplot()+
  geom_sf(data=barrios_caba)+
  geom_sf(data=datos_caba_geo, aes(color=BARRIO), alpha=0.5, show.legend = FALSE)
```



Ya tenemos solo los registros de CABA. ¿Y ahora cómo hacemos si queremos realizar un mapa coroplético como los anteriores pero coloreando los barrios según valor del m2? Nuevamente manipulamos el set de datos con `group_by()` y luego utilizamos `left_join()`:

```
datos_caba_geo <- datos_caba_geo %>%
  group_by(BARRIO, operation_type) %>%
  summarise(valor_m2=mean(price/surface_covered))

head(datos_caba_geo)

## Simple feature collection with 6 features and 3 fields
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:            xmin: -58.50249 ymin: -34.62174 xmax: -58.39217 ymax: -34.58573
## CRS:             EPSG:4326
## # A tibble: 6 x 4
## # Groups:   BARRIO [3]
##   BARRIO  operation_type valor_m2                               geometry
##   <fct>    <fct>        <dbl>                                <MULTIPOLYGON [°]>
## 1 AGRONOM~ Alquiler      471. ((-58.50249 -34.59347), (-58.49966 -34.59564-
## 2 AGRONOM~ Venta         2869. ((-58.49994 -34.59319), (-58.49984 -34.59559-
## 3 ALMAGRO  Alquiler      488. ((-58.43224 -34.60375), (-58.43216 -34.60207-
## 4 ALMAGRO  Venta         2714. ((-58.43277 -34.60322), (-58.43229 -34.60277-
```

```
## 5 BALVANE~ Alquiler      478. ((-58.4128 -34.6028), (-58.41252 -34.60927), ~
## 6 BALVANE~ Venta        2211. ((-58.41409 -34.61113), (-58.41314 -34.61122~
```

Pero como vimos al principio de la clase, si usamos `left_join()`, uno de los dataset no tiene que ser espacial. Para esto, transformemos nuestro dataset espacial `data_caba_geo` a un dataset tradicional, quitándole la geometría con `st_set_geometry()`:

```
datos_caba_geo <- datos_caba_geo %>%
  st_set_geometry(NULL)
```

```
head(datos_caba_geo)
```

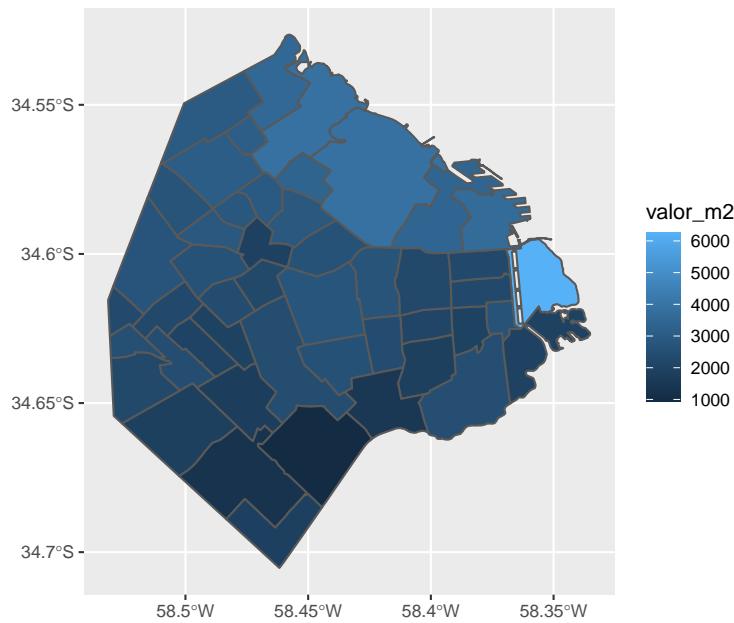
```
## # A tibble: 6 x 3
## # Groups:   BARRIO [3]
##   BARRIO    operation_type valor_m2
##   <fct>    <fct>          <dbl>
## 1 AGRONOMIA Alquiler       471.
## 2 AGRONOMIA Venta         2869.
## 3 ALMAGRO  Alquiler       488.
## 4 ALMAGRO  Venta          2714.
## 5 BALVANERA Alquiler      478.
## 6 BALVANERA Venta         2211.
```

Ahora si, solo tiene 3 columnas con el nombre del barrio, la operación inmobiliaria y el valor del m<sup>2</sup> pero ya no tiene geometría. Procedamos a realizar la unión:

```
barrios_caba <- left_join(barrios_caba, datos_caba_geo, by="BARRIO")
```

Y a mapear el valor de Venta:

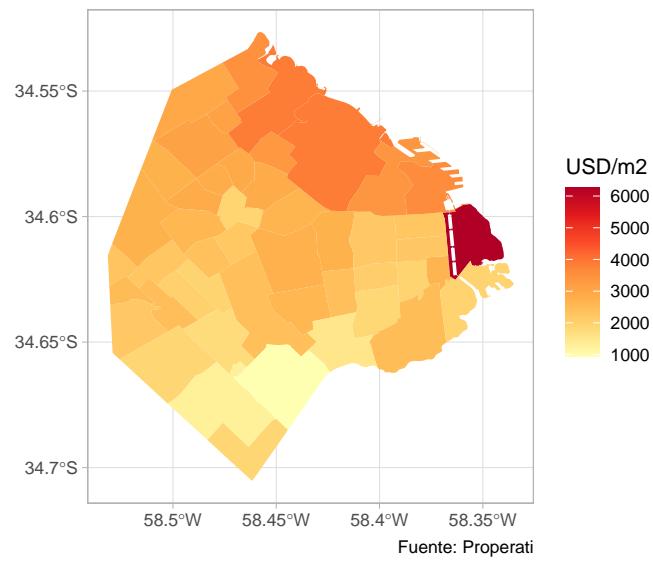
```
ggplot()+
  geom_sf(data=filter(barrios_caba, operation_type=="Venta"), aes(fill=valor_m2))
```



Ya tenemos un mapa de valor del m<sup>2</sup> en Venta por Barrio, y como siempre Puerto Madero presenta el valor más alto. Mejoremos su estética:

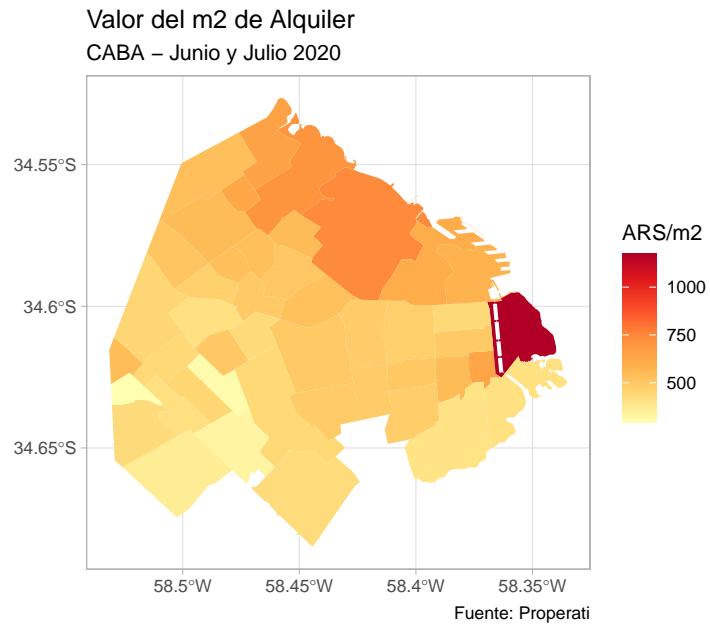
```
ggplot()+
  geom_sf(data=filter(barrios_caba, operation_type=="Venta"), aes(fill=valor_m2), color=NA) +
  labs(title = "Valor del m2 de Venta",
       subtitle = "CABA - Junio y Julio 2020",
       fill = "USD/m2",
       caption= "Fuente: Properati") +
  scale_fill_distiller(palette = "YlOrRd", direction = 1) +
  theme_light()
```

Valor del m<sup>2</sup> de Venta  
CABA – Junio y Julio 2020



Y el de valor del m<sup>2</sup> en alquiler:

```
ggplot()+
  geom_sf(data=filter(barrios_caba, operation_type=="Alquiler"), aes(fill=valor_m2), color=NA)
  labs(title = "Valor del m2 de Alquiler",
       subtitle = "CABA - Junio y Julio 2020",
       fill = "ARS/m2",
       caption= "Fuente: Properati") +
  scale_fill_distiller(palette = "YlOrRd", direction = 1) +
  theme_light()
```



En el mapa anterior se puede observar que hay barrios en el sur de la Ciudad que directamente no tienen propiedades en alquiler.

## 4.4 Agregar mapa base

Si bien hasta aquí hemos hecho mapas bastante completos, nunca está de más agregarles un fondo que nos ayude a interpretar el contexto. Para lograr esto utilizaremos la librería `ggmap()` que como ya sabemos, primero debemos instalarla y luego activarla:

```
#install.packages(ggmap)
library(ggmap)
```

Para poder obtener un mapa de fondo con `ggmap`, primero tengo que delimitar cuál es mi “bounding box”, que hace referencia a los límites de un cuadro en el cual se encuentran todos mis datos. Esto lo haremos de la siguiente forma:

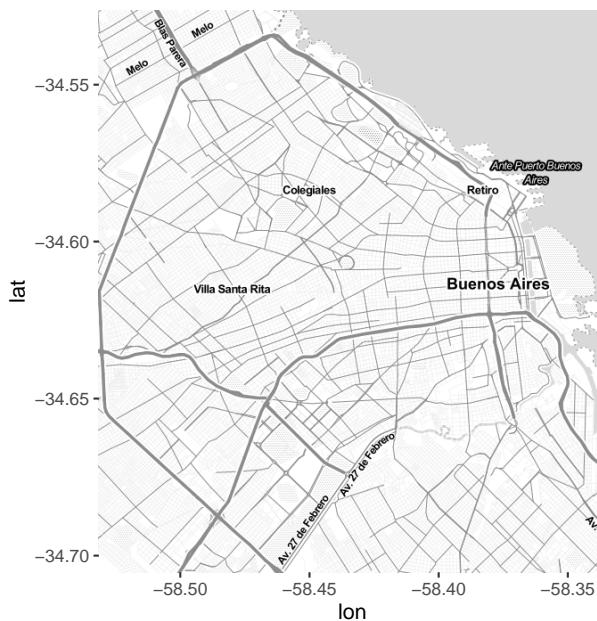
```
bbox_barrios <- as.numeric(st_bbox(barrios_caba))
```

Una vez que ya tenemos el `bbox`, vamos a usar `get_stamenmap()` para descargar de internet el mapa. En este caso utilizaremos un mapa de tipo “toner-lite” pero hay otras opciones que pueden verse aquí: <http://maps.stamen.com/>

```
mapa_caba <- get_stamenmap(bbox = bbox_barrios,
                           maptype = "toner-lite",
                           zoom=12)
```

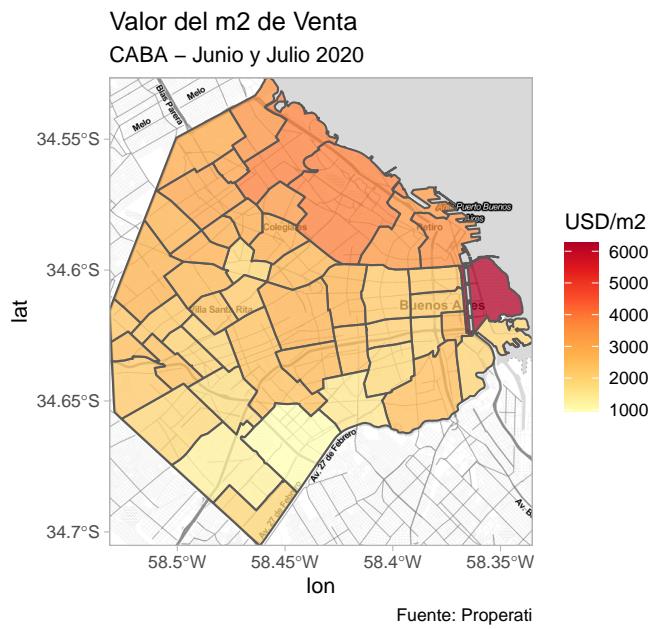
Con `ggmap()` veamos que nos hemos descargado:

```
ggmap(mapa_caba)
```



Ahora podemos reutilizar el código del ejemplo de mapa del valor de m<sup>2</sup> de venta por barrio, simplemente reemplazando la primera línea, donde inicializamos un objeto ggplot por una línea que llame a nuestro mapa base de la siguiente forma:

```
ggmap(mapa_caba) +
  geom_sf(data=filter(barrios_caba, operation_type=="Venta"), aes(fill=valor_m2), alpha=0.6) +
  labs(title = "Valor del m2 de Venta",
       subtitle = "CABA - Junio y Julio 2020",
       fill = "USD/m2",
       caption= "Fuente: Properati") +
  scale_fill_distiller(palette = "YlOrRd", direction = 1) +
  theme_light()
```



Además de cambiar la primer línea de código, nótese que agregamos un inherit.aes=FALSE dentro del geom\_sf() para anular la estética predeterminada del objeto ggmap.

Y listo! Ya tenemos un mapa con fondo y con información que generamos nosotros a partir de un cruce espacial.