1er parcial - Análisis de datos científicos en R

• • Tienen dos horas+ para completar el parcial. • Tienen todo el material de clase y online disponible para consulta. • NO tienen a sus compañeros disponibles. • Tienen que resolver los ejercicios escribiendo las respuestas (tanto texto como código) en un archivo de texto (puede ser una notebook en markdown, evitar usar Word, si esto los complica, avísenme por favor). • Importantísimo: enumerar de la misma manera las respuestas (aunque estén vacías) para que coincidan correctamente. Notar que cada pregunta y subpregunta están numeradas. • Algunas preguntas son solo para los alumnos que cursan la materia como POSGRADO. Para el resto de los alumnos son opcionales, pero suman ;) • Al final del parcial me mandan el archivo de texto o notebook por mail o por mensaje privado en Google Classroom (o en un pendrive si no tienen Internet). • Si no saben o no les sale algo, intenten explicar todo lo que sí saben al respecto, aunque sea parcial o incompleto, porque puede sumarles puntos. Si una línea de código les da algún error, expliquen la lógica de lo que quisieron hacer, ya que si la lógica es correcta, eso también suma. Si identifican la causa del error, aunque no lo sepan corregir, nuevamente eso suma. En general, mejor poner de más que de menos. • No se estresen!

```
library("tidyverse")
```

```
## -- Attaching packages ------------------------------------------------ tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0       v purrr   0.3.1
## v tibble  2.0.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0
```

```
## -- Conflicts --------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

1. DATOS 1.1 Cargar los datos "txhousing" del paquete ggplot2 en la sesión (o verificar si ya se encuentran disponibles). Son datos de venta de inmuebles en Texas, con datos de ciudad, año, mes, número total de inmuebles vendidos (sales), volumen total vendido en dólares (volume), mediana del valor del inmueble, número de inmuebles disponibles (listings) y meses que se demoraría en vender todo lo disponible al ritmo de venta actual (inventory).

```
data("txhousing")
```

1.2. Renombrar el dataframe a "tx" (o sea, asignar el objeto txhousing al objeto tx).

```
tx <- txhousing
```

2.1. ¿Qué clase de estructura de datos es tx?

```
class(tx)
```

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

Es un data.frame (y también un tbl y un tbl_df). También se ve con str().

2.2. ¿Qué dimensiones tiene tx y con que comando lo veo?

```
dim(tx)
```

```
## [1] 8602    9
```

Filas y columnas. También se puede ver directamente de str() o glimpse().

2.3. ¿Cómo son los tipos de datos de cada elemento de tx? Poner la instrucción usada para ver esto.

```
class(tx$city)
```

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

Se puede hacer class() para todas las columnas, o sacarlo después de los dos puntos de str(). Está escrito al lado del nombre de la columna en glimpse(), también.

2.4. *POSGRADO* Usando subsetting o dplyr, guardar en una variable llamada "med_houston_10_2010" la mediana del valor de inmuebles vendidos en octubre de 2010 en Houston. La variable debería contener solamente la mediana, es decir, que "med_houston_10_2010" no debería ser un dataframe.

```
med_houston_10_2010 <- tx %>%
  group_by(month, year) %>%
  summarise(median_sales = median(sales, na.rm = TRUE)) %>%
  filter(month == 10 & year == 2010) %>%
  ungroup() %>%
  select(median_sales) %>%
  as.numeric()

med_houston_10_2010
```

```
## [1] 112
```

3. PRIMER ANÁLISIS - responder usando herramientas de dplyr y/o subsetting. 3.1. ¿Cuántas ciudades diferentes hay en el dataframe tx?

```
tx %>% group_by(city) %>% tally() %>% select(city) %>% nrow()
```

```
## [1] 46
```

Hay 46 ciudades.

Análisis cuantitativo: 3.2.1. Contar cuántas observaciones (filas) hay por ciudad (o sea, debe haber una columna con las ciudades y otra con el número de observaciones de cada ciudad).

```
tx %>% group_by(city) %>% tally()
```

```
## # A tibble: 46 x 2
##    city                     n
##    <chr>                <int>
##  1 Abilene                187
##  2 Amarillo               187
##  3 Arlington              187
##  4 Austin                 187
##  5 Bay Area               187
##  6 Beaumont               187
##  7 Brazoria County        187
##  8 Brownsville            187
##  9 Bryan-College Station  187
## 10 Collin County          187
## # ... with 36 more rows
```

Hay 187 observaciones para cada ciudad.

3.2.2. Contar cuántas observaciones hay por ciudad y año (ahora tendremos ciudad, año, y número de observaciones).

```
tx %>% group_by(city, year) %>% tally()
```

```
## # A tibble: 736 x 3
```

```
## # Groups:   city [46]
##    city     year     n
##    <chr>   <int> <int>
##  1 Abilene  2000    12
##  2 Abilene  2001    12
##  3 Abilene  2002    12
##  4 Abilene  2003    12
##  5 Abilene  2004    12
##  6 Abilene  2005    12
##  7 Abilene  2006    12
##  8 Abilene  2007    12
##  9 Abilene  2008    12
## 10 Abilene  2009    12
## # ... with 726 more rows
```

Hay 12 observaciones para cada ciudad y año, correspondientes a los meses de cada año.

Análisis de precio: Tomando en cuenta todos los años (es decir todo el dataframe): 3.3.1. ¿Cuál ciudad tiene los inmuebles más caros en promedio?

```
tx %>%
  group_by(city) %>%
  summarise(mean_price_per_house = mean(volume/sales, na.rm=T)) %>%
  arrange(desc(mean_price_per_house)) %>%
  head(1)
```

```
## # A tibble: 1 x 2
##   city          mean_price_per_house
##   <chr>                        <dbl>
## 1 Collin County              242186.
```

3.3.2. ¿Cuál ciudad tiene los más baratos en promedio?

```
tx %>%
  group_by(city) %>%
  summarise(mean_price_per_house = mean(volume/sales, na.rm=T)) %>%
  arrange(desc(mean_price_per_house)) %>%
  tail(1)
```

```
## # A tibble: 1 x 2
##   city  mean_price_per_house
##   <chr>                <dbl>
## 1 Paris               99279.
```

Análisis temporal: Construir una nuevo dataframe llamado "tx_by_yr", con el volumen total de ventas y el número total de inmuebles vendidos, por año (es decir, habrá tres columnas: año, total volumen de ventas y total número de inmuebles vendidos).

```
tx_tot_y <- tx %>% group_by(year) %>%
  summarise(tot_vol=sum(volume, na.rm=T), tot_sales=sum(sales, na.rm=T))
```

3.4.1. ¿En cuál año se recaudó más en total?

```
tx_tot_y %>% arrange(desc(tot_vol)) %>% head(n=1)
```

```
## # A tibble: 1 x 3
##    year     tot_vol tot_sales
##   <int>       <dbl>     <dbl>
## 1  2014 84760948831    345720
```

3.4.2. ¿En cuál se vendieron menos inmuebles?

```
tx_tot_y %>% arrange(tot_sales) %>% head(n=1)
```

```
## # A tibble: 1 x 3
##    year     tot_vol tot_sales
##   <int>       <dbl>     <dbl>
## 1  2015 54118881305    208124
```

3.5.*POSGRADO* ¿En que mes y año se vendieron más cantidad de inmuebles (en número, o sea, sales) en la ciudad de "Amarillo"?

```
tx %>%
  filter(city == "Amarillo") %>%
  group_by(year, month) %>%
  summarise(tot_sales=sum(sales, na.rm=T)) %>%
  arrange(desc(tot_sales)) %>%
  head(1)
```

```
## # A tibble: 1 x 3
## # Groups:   year [1]
##    year month tot_sales
##   <int> <int>     <dbl>
## 1  2011     8       390
```

3.6. *POSGRADO* Agregar al dataframe original una columna adicional llamada frac, que tenga el número total de inmuebles vendidos (sales) dividido por el de disponibles (listings). Hacer un promedio de frac para cada año.

```
tx %>%
  mutate(frac=sales/listings) %>%
  group_by(year) %>%
  summarise(frac_prom = mean(frac, na.rm = T))
```

```
## # A tibble: 16 x 2
##     year frac_prom
##    <int>     <dbl>
##  1  2000     0.199
##  2  2001     0.186
##  3  2002     0.182
##  4  2003     0.171
##  5  2004     0.171
##  6  2005     0.191
##  7  2006     0.202
##  8  2007     0.181
##  9  2008     0.146
## 10  2009     0.135
## 11  2010   Inf
## 12  2011     0.128
## 13  2012     0.175
## 14  2013     0.234
## 15  2014     0.255
## 16  2015     0.286
```

4. GRAFICAR

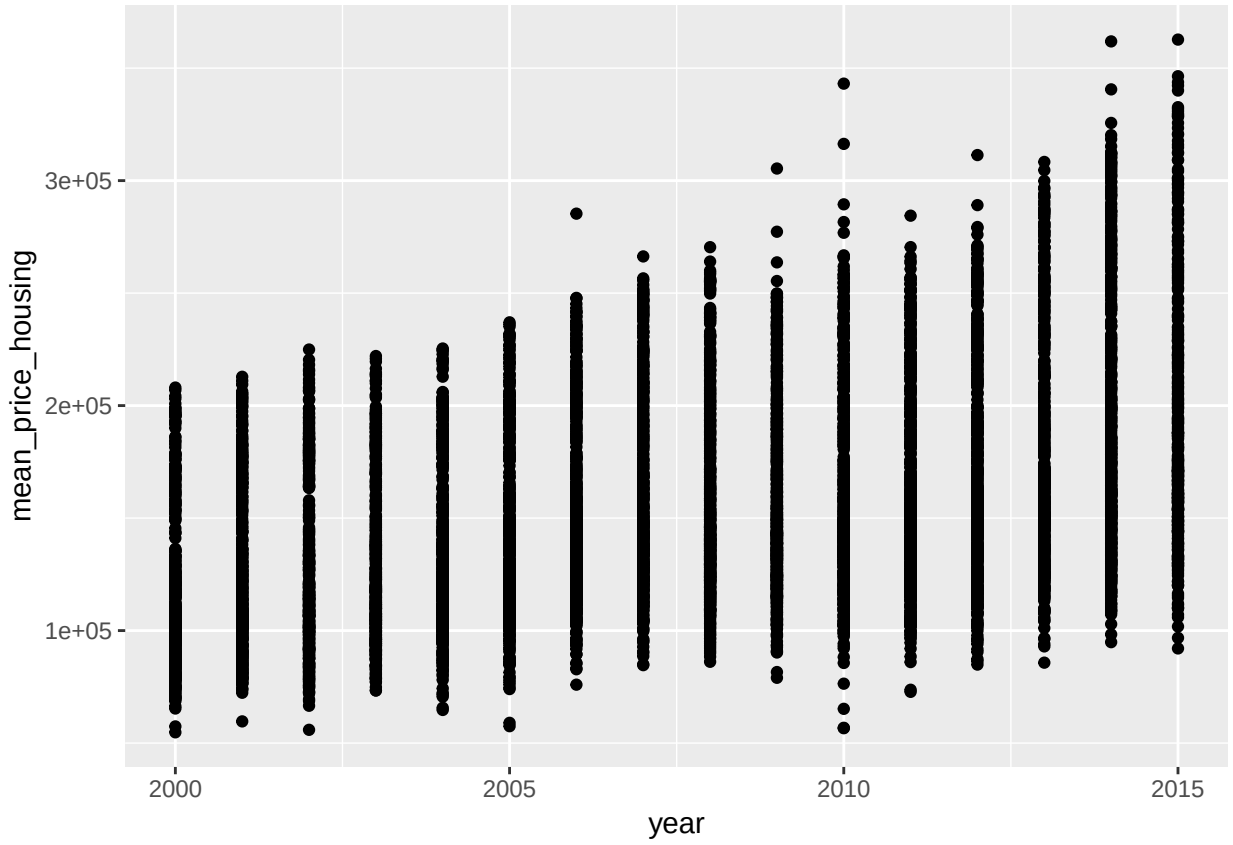
Este punto utiliza el dataframe creado en el punto 3.4. Si no salió por alguna razón, pueden crear columnas artificiales usando las herramientas de R para generar números aleatorios, como rnorm() o runif(). Si se dan

maña para hacer esto, se considerará el ejercicio (casi) resuelto ;)

4.1. Graficar, con puntos, el volumen total de ventas dividido el número total de inmuebles vendidos, en función del año.

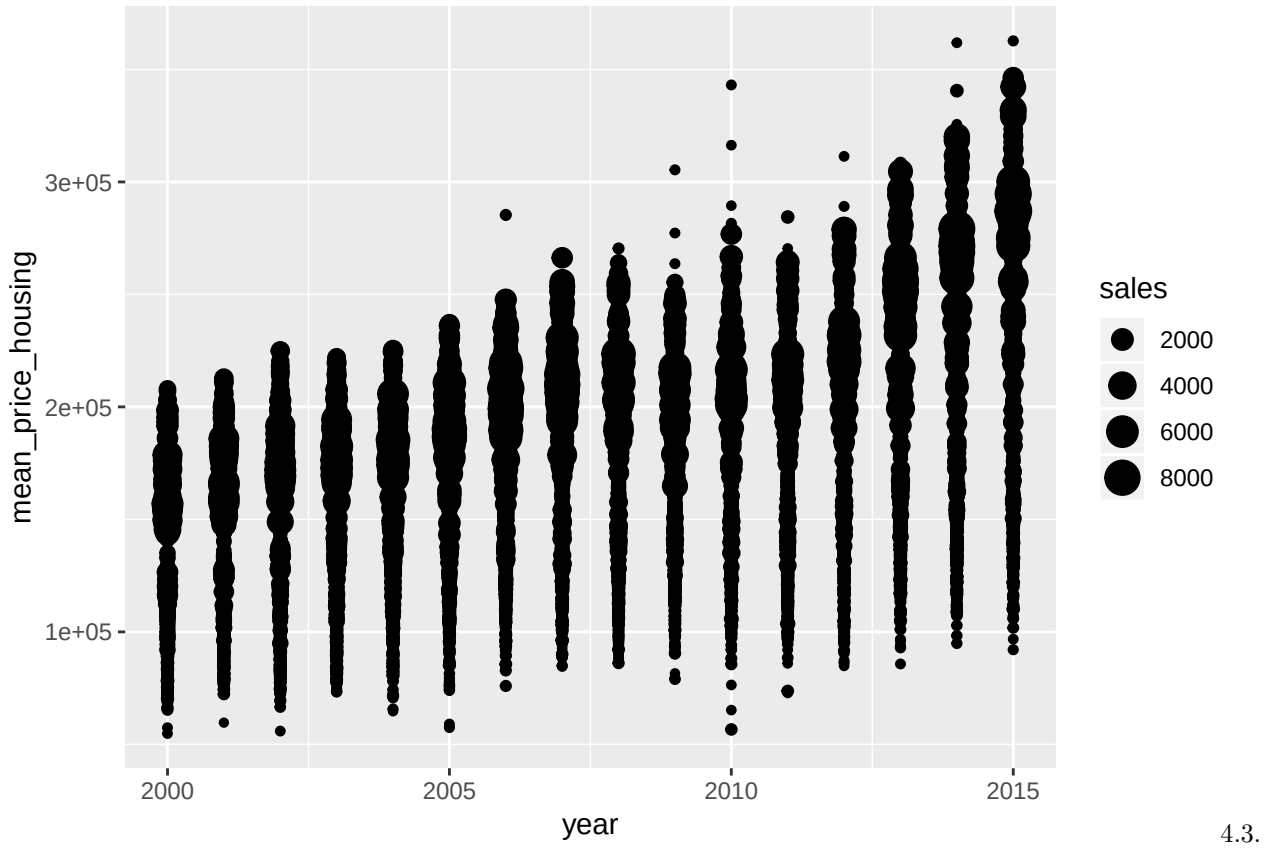
Los que resolvieron los siguientes ejercicios agrupando por año y tomando promedios o sumas, no está mal. No es estrictamente lo que se pide, pero la diferencia es menor.

```
tx_y <- tx %>% mutate( mean_price_housing = volume/sales) %>% filter(!is.na(mean_price_housing))
ggplot(tx_y) + geom_point(aes(year, mean_price_housing))
```



4.2. Mapear el tamaño de los puntos al número total de inmuebles vendidos ese año (puntos más grandes significarán mayor número de inmuebles).

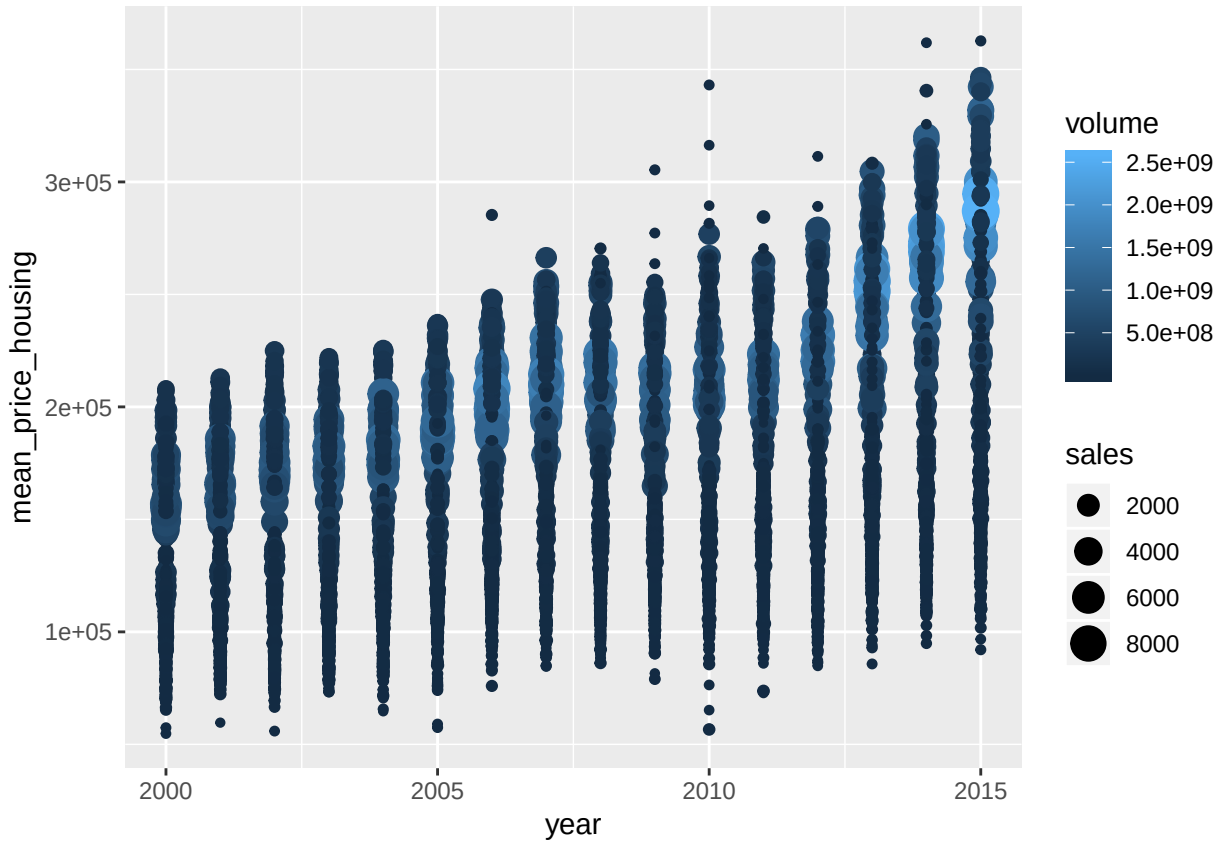
```
ggplot(tx_y) + geom_point(aes(year, mean_price_housing, size = sales))
```



4.3.

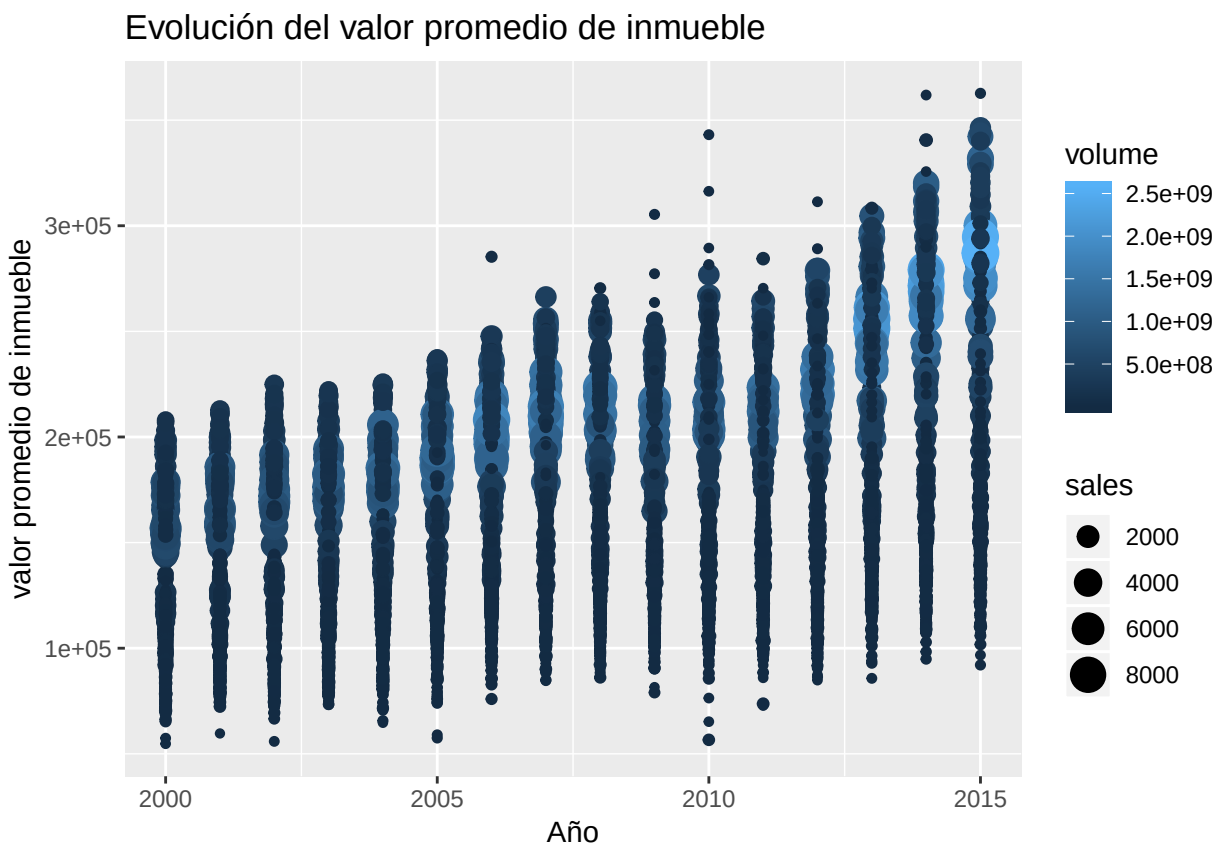
Mapear el color de los puntos al volumen total recaudado.

```
ggplot(tx_y) + geom_point(aes(year, mean_price_housing, size = sales, color = volume))
```



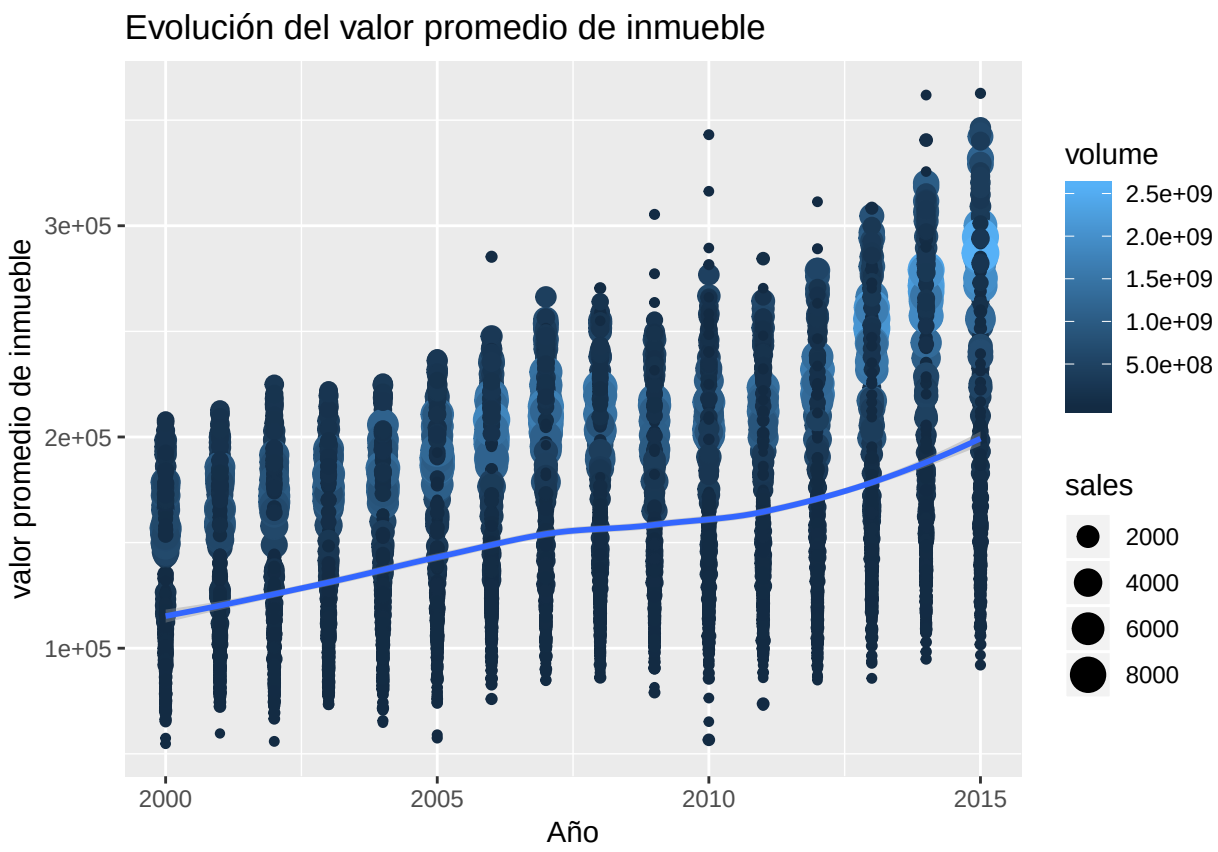
4.4. Ponerle texto explicativo a los ejes x e y, y título al gráfico.

```
ggplot(tx_y) +
  geom_point(aes(year, mean_price_housing, size = sales, color = volume)) +
  labs(x = "Año") +
  labs(y = "valor promedio de inmueble") +
  labs(title = "Evolución del valor promedio de inmueble")
```



4.5.*POSGRADO* Agregar una regresión de tipo "loess" usando una capa geom_smooth().

```
ggplot(tx_y) +
  geom_point(aes(year, mean_price_housing, size = sales, color = volume)) +
  labs(x = "Año") +
  labs(y = "valor promedio de inmueble") +
  labs(title = "Evolución del valor promedio de inmueble") +
  geom_smooth( aes(x = year, y = mean_price_housing), method = "loess")
```


Evolución del valor promedio de inmueble
valor promedio de inmueble
3e+05
2e+05
1e+05
2000
2005
2010
2015
Año
volume
2.5e+09
2.0e+09
1.5e+09
1.0e+09
5.0e+08
sales
2000
4000
6000
8000