Práctica Calidad del dato

Fernando Gualo

Descripción de Librerías en R

Esta sección contiene una breve descripción del propósito de las librerías instaladas en R y para qué se utilizan en el análisis de datos.

Librerías para Conexión y Manejo de Bases de Datos

`DBI`

- **Descripción**: Proporciona una interfaz genérica para interactuar con bases de datos desde R.
- **Uso**: Utilizado como capa base para conectar con diferentes motores de bases de datos (e.g., MySQL, PostgreSQL).
- **Funciones clave**: `dbConnect()`, `dbGetQuery()`, `dbWriteTable()`, `dbDisconnect()`.

`RMySQL`

- **Descripción**: Paquete específico para la conexión y manipulación de bases de datos MySQL desde R.
- **Uso**: Permite ejecutar consultas SQL, extraer datos y realizar operaciones dentro de bases de datos MySQL.
- **Funciones clave**: `dbConnect(MySQL())`, `dbSendQuery()`, `fetch()`, `dbDisconnect()`.

Librerías para Lectura de Datos

`readxl`

- **Descripción**: Permite la lectura de archivos Excel (.xls y .xlsx) en R sin necesidad de depender de software externo.
- **Uso**: Muy útil para importar datos en formato Excel.
- **Funciones clave**: `read_excel()`, `excel_sheets()`.

Librerías para Manipulación y Transformación de Datos

`dplyr`

- **Descripción**: Parte del ecosistema `tidyverse`, facilita la manipulación de dataframes con una sintaxis clara y concisa.
- **Uso**: Útil para seleccionar, filtrar, agrupar y modificar datos de manera eficiente.
- **Funciones clave**: `select()`, `filter()`, `mutate()`, `summarise()`, `group_by()`.

`tidyr`

- **Descripción**: Parte del ecosistema `tidyverse` y se usa para transformar la estructura de los dataframes.
- **Uso**: Facilita la conversión entre formatos anchos y largos, y la reorganización de datos con funciones como `pivot_longer()` y `pivot_wider()`.
- **Funciones clave**: `gather()`, `spread()`, `pivot_longer()`, `pivot_wider()`.

Librerías para Análisis de Calidad de Datos

`naniar`

- **Descripción**: Enfocada en la visualización y manejo de datos faltantes.
- **Uso**: Proporciona funciones para explorar la presencia de valores NA y patrones de datos faltantes.
- **Funciones clave**: `gg_miss_var()`, `vis_miss()`, `replace_with_na()`.

`visdat`

- **Descripción**: Facilita la creación de visualizaciones de calidad de datos, como patrones de valores faltantes y estructuras de datos.
- **Uso**: Útil para obtener una visión general de la estructura y calidad de un dataframe.
- **Funciones clave**: `vis_dat()`, `vis_miss()`, `vis_cor()`.

Librerías para Visualización de Datos

`ggplot2`

- **Descripción**: Paquete de visualización de datos basado en la gramática de gráficos.
- **Uso**: Permite crear gráficos complejos de manera sencilla y altamente personalizable.
- **Funciones clave**: `ggplot()`, `geom_point()`, `geom_line()`, `facet_wrap()`.

Librerías para Visualización de Correlaciones y Análisis Estadístico

`corrplot`

- **Descripción**: Permite crear gráficos de matrices de correlación de manera fácil y visual.
- **Uso**: Muy útil para identificar relaciones entre variables numéricas.
- **Funciones clave**: `corrplot()`, `corrgram()`.

---

1. Cargar Librerías y Conjunto de Datos

En primer lugar, será necesario realizar una carga de los datos en un frame de R, para ello en este caso Cargaremos todos los datos provenientes de un fichero Excel. Los datos son los mismos que los que se utilizan para el ejercicio propuesto con la diferencia que para el propuesto la conexión será necesaria a la base de datos relacional.

```r
# Cargar librerías
library(DBI)
library(RMySQL)
library(readxl)
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)
library(naniar)
library(visdat)
library(ggplot2)
library(tidyr)
library(corrplot)
```

```
## corrplot 0.94 loaded
```

```r
# Cargar el archivo de datos
data <- read_excel("C:/Users/Fernando.Gualo/Downloads/housing_price_madrid_v1.1.xlsx", sheet = "viviend

# Ver las primeras filas para confirmar la carga
head(data)
```

```
## # A tibble: 6 x 52
##    Obs. barrio   cod_barrio distrito cod_distrito longitud latitud      X      Y
##   <dbl> <chr>         <dbl> <chr>           <dbl>    <dbl>   <dbl>  <dbl>  <dbl>
## 1   595 Imperial         21 arganzu~            2    -3.72    40.4 4.39e5 4.47e6
## 2   219 Imperial         21 arganzu~            2    -3.72    40.4 4.39e5 4.47e6
## 3   745 Imperial         21 arganzu~            2    -3.72    40.4 4.39e5 4.47e6
## 4   555 Imperial         21 arganzu~            2    -3.72    40.4 4.39e5 4.47e6
## 5    77 Imperial         21 arganzu~            2    -3.72    40.4 4.39e5 4.47e6
## 6   712 Imperial         21 arganzu~            2    -3.71    40.4 4.40e5 4.47e6
## # i 43 more variables: precio.house.em2 <dbl>, sup.const <dbl>, sup.util <dbl>,
## #   ref.hip.zona <dbl>, hipot.mens <dbl>, dorm <dbl>, banos <dbl>,
## #   tipo.house. <chr>, planta.house. <chr>, inter.exter. <chr>,
## #   ascensor. <chr>, aire.acond. <chr>, armar.emp. <chr>, agua.cal. <chr>,
## #   calef. <chr>, cocina. <chr>, tendedero. <chr>, orientac. <chr>,
## #   estado. <chr>, trastero. <chr>, piscina. <chr>, garaje. <chr>, antig <dbl>,
## #   antig.interv <chr>, comercial <dbl>, casco.historico <dbl>, ...
```

2. Análisis Exploratorio de la Estructura

En segundo lugar, tenemos que conocer los datos, por lo que se realizará un análisis exploratorio a distintos niveles: conocer las dimensiones que tiene el dataset, tipos de datos, etc.

2.1. Dimensiones del Conjunto de Datos

A continuación se puede ver información general sobre el volumen de registros y la cantidad de dimensiones o propiedades dentro del dataset.

```
# Dimensiones de cada una de las columnas
dim(data)
```

```
## [1] 1000   52
```

2.2. Tipos de Datos

A través de esta visualización sencilla se pude tener una idea de que contiene cada uno de las dimensiones así como su tipología.

```
# Mostrar los tipos de datos de cada columna
str(data)
```

```
## tibble [1,000 x 52] (S3: tbl_df/tbl/data.frame)
##  $ Obs.                          : num [1:1000] 595 219 745 555 77 ...
##  $ barrio                        : chr [1:1000] "Imperial" "Imperial" "Imperial" "Imperial" ...
##  $ cod_barrio                    : num [1:1000] 21 21 21 21 21 21 21 21 22 22 ...
##  $ distrito                      : chr [1:1000] "arganzuela" "arganzuela" "arganzuela" "arganzuela" .
##  $ cod_distrito                  : num [1:1000] 2 2 2 2 2 2 2 2 2 2 ...
##  $ longitud                      : num [1:1000] -3.72 -3.72 -3.72 -3.72 -3.72 ...
##  $ latitud                       : num [1:1000] 40.4 40.4 40.4 40.4 40.4 ...
##  $ X                             : num [1:1000] 438850 438847 439054 439255 438974 ...
##  $ Y                             : num [1:1000] 4473641 4473767 4473177 4473101 4473014 ...
##  $ precio.house.em2              : num [1:1000] 4800 3356 4000 4592 3303 ...
##  $ sup.const                     : num [1:1000] 75 73 35 65 47 75 55 78 65 60 ...
##  $ sup.util                      : num [1:1000] 60 59 29 46 40 69 48 57 55 53 ...
##  $ ref.hip.zona                  : num [1:1000] 3410 3524 4082 3928 3812 ...
##  $ hipot.mens                    : num [1:1000] 1372 934 534 1138 592 ...
##  $ dorm                          : num [1:1000] 3 2 1 1 2 2 3 3 2 1 ...
##  $ banos                         : num [1:1000] 1 1 1 1 1 1 1 1 1 1 ...
##  $ tipo.house.                   : chr [1:1000] "piso" "piso" "piso" "piso" ...
##  $ planta.house.                 : chr [1:1000] "septima" "primera" NA "cuarta" ...
##  $ inter.exter.                  : chr [1:1000] "exterior" "interior" "interior" "exterior" ...
##  $ ascensor.                     : chr [1:1000] "si" "Sí" "Sí" "Sí" ...
##  $ aire.acond.                   : chr [1:1000] "no" "si" "no" "si" ...
##  $ armar.emp.                    : chr [1:1000] "no" "si" "no" "si" ...
##  $ agua.cal.                     : chr [1:1000] "Electrica" "Individual" "Gas_Butano" "Gas_Natural" .
##  $ calef.                        : chr [1:1000] "Centr_Com." "Individual" "Electrica" "Gas_Natural" .
##  $ cocina.                       : chr [1:1000] "si" "si" "si" "si" ...
##  $ tendedero.                    : chr [1:1000] "no" "si" "no" "si" ...
##  $ orientac.                     : chr [1:1000] "norte" NA "este" "2_o_mas" ...
##  $ estado.                       : chr [1:1000] "buen_estado" "reformado" "buen_estado" "buen_estado"
##  $ trastero.                     : chr [1:1000] "No" "no" "no" "no" ...
##  $ piscina.                      : chr [1:1000] "no" "no" "no" "si" ...
##  $ garaje.                       : chr [1:1000] "no" "no" "no" "no" ...
##  $ antig                         : num [1:1000] 25 30 22.3 2.5 40 ...
##  $ antig.interv                  : chr [1:1000] "(18.9,27]" "(27,34.8]" "(18.9,27]" "[0,9.08]" ...
```

```
##  $ comercial                    : num [1:1000] 1 1 1 1 1 1 1 0 1 1 ...
##  $ casco.historico              : num [1:1000] 1 1 1 1 1 1 1 0 1 1 ...
##  $ transp.publ.                 : chr [1:1000] "bueno" "bueno" "malo" "bueno" ...
##  $ Ruidos_ext                   : num [1:1000] 0.39 0.39 0.39 0.39 0.39 0.39 0.39 0.39 0.42 0.42 ...
##  $ Mal_olor                     : num [1:1000] 0.31 0.31 0.31 0.31 0.31 0.31 0.31 0.31 0.35 0.35 ...
##  $ Poca_limp                    : num [1:1000] 0.49 0.49 0.49 0.49 0.49 0.49 0.49 0.49 0.48 0.48 ...
##  $ Malas_comunic                : num [1:1000] 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.22 0.14 0.14 ...
##  $ Pocas_zonas                  : num [1:1000] 0.28 0.28 0.28 0.28 0.28 0.28 0.28 0.28 0.24 0.24 ...
##  $ Delincuencia                 : num [1:1000] 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.33 0.4 0.4 ...
##  $ M.30                         : num [1:1000] 0 0 0 0 0 0 0 0 0 0 ...
##  $ CO                           : num [1:1000] 0.0321 0.0019 0.2068 0.267 0.2485 ...
##  $ NO2                          : num [1:1000] 0.192 0.122 0.628 0.932 0.47 ...
##  $ Nox                          : num [1:1000] 0.106 0.067 0.35 0.521 0.262 ...
##  $ O3                           : num [1:1000] 0.787 0.724 1.126 1.328 1.015 ...
##  $ SO2                          : num [1:1000] -0.373 -0.356 -0.441 -0.503 -0.413 ...
##  $ PM10                         : num [1:1000] 0.0849 0.0857 0.1167 0.1574 0.0965 ...
##  $ Pobl.0_14_div_Poblac.Total   : num [1:1000] 12.4 12.4 12.4 12.4 12.4 ...
##  $ PoblJubilada_div_Poblac.Total: num [1:1000] 16.5 16.5 16.5 16.5 16.5 ...
##  $ Inmigrantes.porc             : num [1:1000] 12.1 12.1 12.1 12.1 12.1 ...
```

2.3. Valores Faltantes

A continuación, se puede ver que el nivel de datos faltantes en el dataset es de 0, aspecto que revisaremos más delante.

```
# Mostrar el número de valores faltantes por cada columna
missing_values <- colSums(is.na(data))
missing_values[missing_values > 0]
```

```
##   tipo.house. planta.house.  inter.exter.     ascensor.   aire.acond.
##            15           195            34            11           117
##    armar.emp.     agua.cal.        calef.       cocina.    tendedero.
##           242           186           116            60           163
##     orientac.       estado.     trastero.      piscina.       garaje.
##           510             6           134           171           183
##  transp.publ.
##            72
```

2.4. Estadísticas Descriptivas

Para ir entrando un poco más en detalle del conjunto de datos, utilizaremos estadisticos descriptivos sobre cad una de las dimensiones de nuestro dataset para ver aspectos como los valores máximo y mínimo, media, o mediana y así tener una aproximación alto nivel de como los datos están distribuidos.

```
# Generar estadísticas descriptivas para todas las columnas numéricas
summary(data)
```

```
##       Obs.          barrio            cod_barrio      distrito
##  Min.   :    2   Length:1000        Min.   : 11.0   Length:1000
##  1st Qu.: 3243   Class :character   1st Qu.: 45.0   Class :character
##  Median : 6607   Mode  :character   Median : 94.5   Mode  :character
##  Mean   : 6513                      Mean   : 96.2
```

```
##  3rd Qu.: 9680                     3rd Qu.:136.0
##  Max.   :12580                     Max.   :215.0
##   cod_distrito       longitud         latitud            X
##  Min.   : 1.000   Min.   :-3.789   Min.   :40.34   Min.   :433055
##  1st Qu.: 4.000   1st Qu.:-3.708   1st Qu.:40.39   1st Qu.:439901
##  Median : 9.000   Median :-3.694   Median :40.42   Median :441176
##  Mean   : 9.285   Mean   :-3.686   Mean   :40.42   Mean   :441765
##  3rd Qu.:13.000   3rd Qu.:-3.664   3rd Qu.:40.45   3rd Qu.:443648
##  Max.   :21.000   Max.   :-3.555   Max.   :40.53   Max.   :452934
##        Y           precio.house.em2   sup.const         sup.util
##  Min.   :4465986   Min.   : 1260    Min.   : 20.00   Min.   : 18.00
##  1st Qu.:4471797   1st Qu.: 2804    1st Qu.: 62.00   1st Qu.: 54.75
##  Median :4474211   Median : 3516    Median : 80.00   Median : 70.00
##  Mean   :4474649   Mean   : 3720    Mean   : 96.54   Mean   : 83.13
##  3rd Qu.:4477556   3rd Qu.: 4432    3rd Qu.:105.00   3rd Qu.: 90.00
##  Max.   :4487147   Max.   :12376    Max.   :875.00   Max.   :680.00
##   ref.hip.zona    hipot.mens        dorm            banos
##  Min.   :1031   Min.   :  80   Min.   :0.000   Min.   :1.000
##  1st Qu.:2824   1st Qu.: 746   1st Qu.:2.000   1st Qu.:1.000
##  Median :3312   Median :1059   Median :2.000   Median :1.000
##  Mean   :3444   Mean   :1451   Mean   :2.526   Mean   :1.505
##  3rd Qu.:3961   3rd Qu.:1622   3rd Qu.:3.000   3rd Qu.:2.000
##  Max.   :6273   Max.   :9149   Max.   :7.000   Max.   :7.000
##  tipo.house.        planta.house.      inter.exter.        ascensor.
##  Length:1000        Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  aire.acond.         armar.emp.         agua.cal.            calef.
##  Length:1000        Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    cocina.           tendedero.         orientac.           estado.
##  Length:1000        Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   trastero.           piscina.           garaje.              antig
##  Length:1000        Length:1000        Length:1000        Min.   : 0.00
##  Class :character   Class :character   Class :character   1st Qu.:15.00
##  Mode  :character   Mode  :character   Mode  :character   Median :29.39
##                                                           Mean   :24.33
##                                                           3rd Qu.:30.00
##                                                           Max.   :75.00
##  antig.interv         comercial    casco.historico transp.publ.
##  Length:1000        Min.   :0.00   Min.   :0.000   Length:1000
##  Class :character   1st Qu.:0.00   1st Qu.:0.000   Class :character
```

```
##  Mode  :character   Median :0.00   Median :0.000   Mode  :character
##                     Mean   :0.41   Mean   :0.338
##                     3rd Qu.:1.00   3rd Qu.:1.000
##                     Max.   :1.00   Max.   :1.000
##    Ruidos_ext        Mal_olor        Poca_limp      Malas_comunic
##  Min.   :0.1200   Min.   :0.0600   Min.   :0.2100   Min.   :0.0100
##  1st Qu.:0.3600   1st Qu.:0.2200   1st Qu.:0.3500   1st Qu.:0.0400
##  Median :0.4000   Median :0.2600   Median :0.4500   Median :0.1200
##  Mean   :0.3978   Mean   :0.2823   Mean   :0.4487   Mean   :0.1579
##  3rd Qu.:0.4300   3rd Qu.:0.3300   3rd Qu.:0.5200   3rd Qu.:0.2100
##  Max.   :0.6400   Max.   :0.7200   Max.   :0.7800   Max.   :0.7200
##   Pocas_zonas      Delincuencia         M.30             CO
##  Min.   :0.0400   Min.   :0.0600   Min.   :0.000   Min.   :-1.42508
##  1st Qu.:0.2200   1st Qu.:0.3000   1st Qu.:0.000   1st Qu.:-0.27164
##  Median :0.3500   Median :0.4300   Median :0.000   Median :-0.02703
##  Mean   :0.3509   Mean   :0.4423   Mean   :0.133   Mean   : 0.05712
##  3rd Qu.:0.4800   3rd Qu.:0.5600   3rd Qu.:0.000   3rd Qu.: 0.42908
##  Max.   :0.7400   Max.   :0.7700   Max.   :1.000   Max.   : 1.60465
##       NO2                 Nox                 O3               SO2
##  Min.   :-1.661229   Min.   :-1.371803   Min.   :-1.20211   Min.   :-1.1641
##  1st Qu.:-0.008158   1st Qu.:-0.003661   1st Qu.:-0.05440   1st Qu.:-0.5863
##  Median :-0.003468   Median :-0.003661   Median : 0.05585   Median :-0.3731
##  Mean   :-0.026384   Mean   : 0.003291   Mean   : 0.07658   Mean   :-0.1530
##  3rd Qu.:-0.002965   3rd Qu.: 0.012695   3rd Qu.: 0.19859   3rd Qu.: 0.1778
##  Max.   : 1.012633   Max.   : 1.602042   Max.   : 1.36501   Max.   : 1.7075
##       PM10          Pobl.0_14_div_Poblac.Total PoblJubilada_div_Poblac.Total
##  Min.   :-0.45637   Min.   : 8.897             Min.   :11.58
##  1st Qu.:-0.24179   1st Qu.:11.434             1st Qu.:16.56
##  Median :-0.06185   Median :12.640             Median :18.68
##  Mean   :-0.01781   Mean   :12.911             Mean   :18.63
##  3rd Qu.: 0.20808   3rd Qu.:14.239             3rd Qu.:20.78
##  Max.   : 0.59965   Max.   :18.374             Max.   :22.75
##  Inmigrantes.porc
##  Min.   : 6.064
##  1st Qu.: 8.575
##  Median :13.431
##  Mean   :13.216
##  3rd Qu.:17.560
##  Max.   :20.543
```

3. Distribución y revisión de Variables

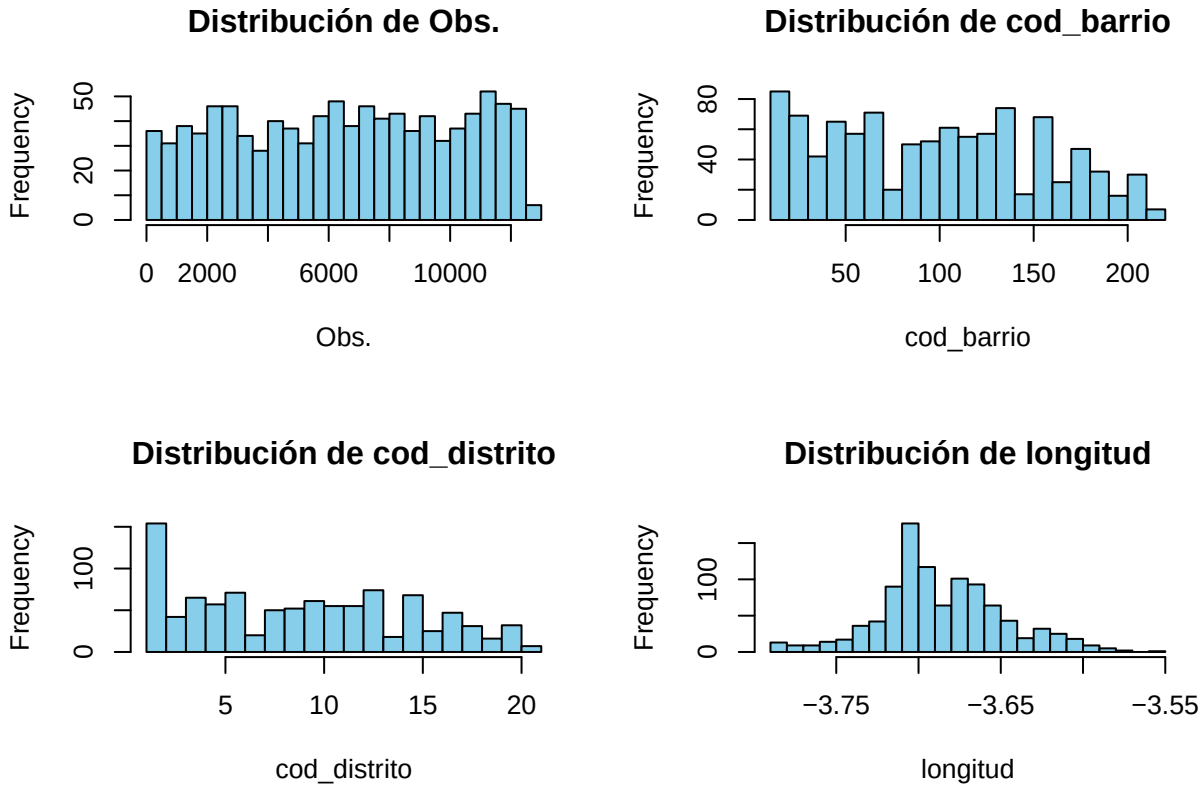
Una vez se dispone del conocimeinto de la estructura básica del dataset, vamos a poner el foco más en detalle en estudiar las variables numéricas y categóricas.

3.1. Distribución de Variables Numéricas

Para el caso de las variables numéricas vamos a utilizar histogramas para determinar como es la distribución de datos.

```
# Distribución de variables numéricas clave (precios, superficies)
numeric_cols <- select_if(data, is.numeric)
```

```r
# Histograma para revisar la distribución de las principales variables numéricas
par(mfrow = c(2, 2))  # Crear una cuadrícula de gráficos
for (col in colnames(numeric_cols)[1:4]) {
  hist(numeric_cols[[col]], main = paste("Distribución de", col), xlab = col, col = "skyblue", breaks =
}
```



3.2. Revisión de Variables Categóricas

Para la revisión de las variables categóricas veremos por ejemplo la distribución del número de barrios y distritos.

```r
# Mostrar frecuencias de variables categóricas como barrio y distrito.
table(data$barrio)
```

```
##
##                                  Abrantes
##                                         6
##                                   Acacias
##                                        21
##                                   Adelfas
##                                        12
##                                Aeropuerto
##                                         1
##                                   Aguilas
```

```
##                                     8
##                                Almenara
##                                     2
##                             Almendrales
##                                     6
##                                 Aluche
##                                    11
##                                Amposta
##                                     1
##                           Apostol Santiago
##                                     3
##                                Arapiles
##                                     2
##                                Aravaca
##                                     7
##                                  Arcos
##                                     7
##                               Argüelles
##                                     7
##                                Atalaya
##                                     1
##                                 Atocha
##                                     4
##                           Bellas Vistas
##                                    16
##                              Berruguete
##                                    13
##                              Buenavista
##                                     8
##                                Butarque
##                                     4
##                              Campamento
##                                     6
##                                Canillas
##                                     3
##                              Canillejas
##                                     9
##                                Cármenes
##                                    11
##                           Casa de Campo
##                                    10
##                 Casco Histórico de Vallecas
##                                    25
##               Casco Histórico de Vicálvaro
##                                    16
##                           Castellana
##                                     5
##                                Castilla
##                                    10
##                             Castillejos
##                                    10
##                                Chopera
##                                    13
##                             Ciudad Jardín
```

```
##                                5
##              Ciudad Universitaria
##                               20
##                         Comillas
##                               10
##                       Concepción
##                                9
##                       Corralejos
##                                2
##                           Cortes
##                                5
##                      Costillares
##                                6
##                    Cuatro Caminos
##                               14
##                    Cuatro Vientos
##                                2
##                          Delicias
##                               14
##                        El Goloso
##                                3
##                         El Pardo
##                                1
##                          El Viso
##                                6
##                      Embajadores
##                               34
##                        Entrevías
##                               15
##                         Estrella
##                                7
##                       Fontarrón
##                                5
##                  Fuente del Berro
##                               12
##                      Gaztambide
##                                2
##                             Goya
##                               13
##                      Guindalera
##                               14
##                          Hellín
##                                2
##                   Hispanoamérica
##                               15
##                         Horcajo
##                                3
##                           Ibiza
##                                6
##                        Imperial
##                                8
##                       Jerónimos
##                                5
##                         Justicia
```

```
##                                     6
##                                La Paz
##                                     4
##                               Legazpi
##                                     4
##                                 Lista
##                                    12
##                           Los Angeles
##                                    10
##                           Los Rosales
##                                    11
##                                Lucero
##                                    11
##                             Marroquina
##                                     4
##                           Media Legua
##                                     1
##                             Mirasierra
##                                     9
##                               Moscardó
##                                     7
##                             Niño Jesús
##                                     2
##                          Nueva España
##                                     8
##                              Numancia
##                                    12
##                                Opañel
##                                     6
##                             Orcasitas
##                                     5
##                               Orcasur
##                                     2
##                              Pacífico
##                                    10
##                               Palacio
##                                     9
##                               Palomas
##                                     1
##                        Palomeras Bajas
##                                    11
##                      Palomeras Sureste
##                                     4
##                        Palos de Moguer
##                                     5
##                                Pavones
##                                     3
##                             Peñagrande
##                                     5
##                                  Pilar
##                                     7
##                          Pinar del Rey
##                                     4
##                                Piovera
```

```
##                                   4
##                           Portazgo
##                                  10
##                          Pradolongo
##                                  17
##                         Prosperidad
##                                  13
##                        Pueblo Nuevo
##                                  17
##                       Puerta Bonita
##                                   7
##                     Puerta del Angel
##                                  12
##                            Quintana
##                                   4
##                           Recoletos
##                                   9
##                               Rejas
##                                   4
##                           Rios Rosas
##                                   5
##                               Rosas
##                                   5
##                            Salvador
##                                   2
##                        San Cristobal
##                                   1
##                            San Diego
##                                  22
##                           San Fermín
##                                   8
##                           San Isidro
##                                  14
##                    San Juan Bautista
##                                   3
##                          San Pascual
##                                   5
##                        Santa Eugenia
##                                   6
##                            Simancas
##                                   2
##                                 Sol
##                                   9
##                               Timón
##                                   4
##                           Trafalgar
##                                   7
##                          Universidad
##                                  22
##                        Valdeacederas
##                                  16
##                         Valdefuentes
##                                  10
##                           Valdemarín
```

```
##                                           4
##                                  Valdezarza
##                                           4
##                                Vallehermoso
##                                           4
##                                   Valverde
##                                          21
##                                     Ventas
##                                          23
## Villaverde Alto, Casco Histórico de Villaverde
##                                          21
##                                   Vinateros
##                                           2
##                                Vista Alegre
##                                           4
##                                      Zofío
##                                          10
```

```
table(data$distrito)
```

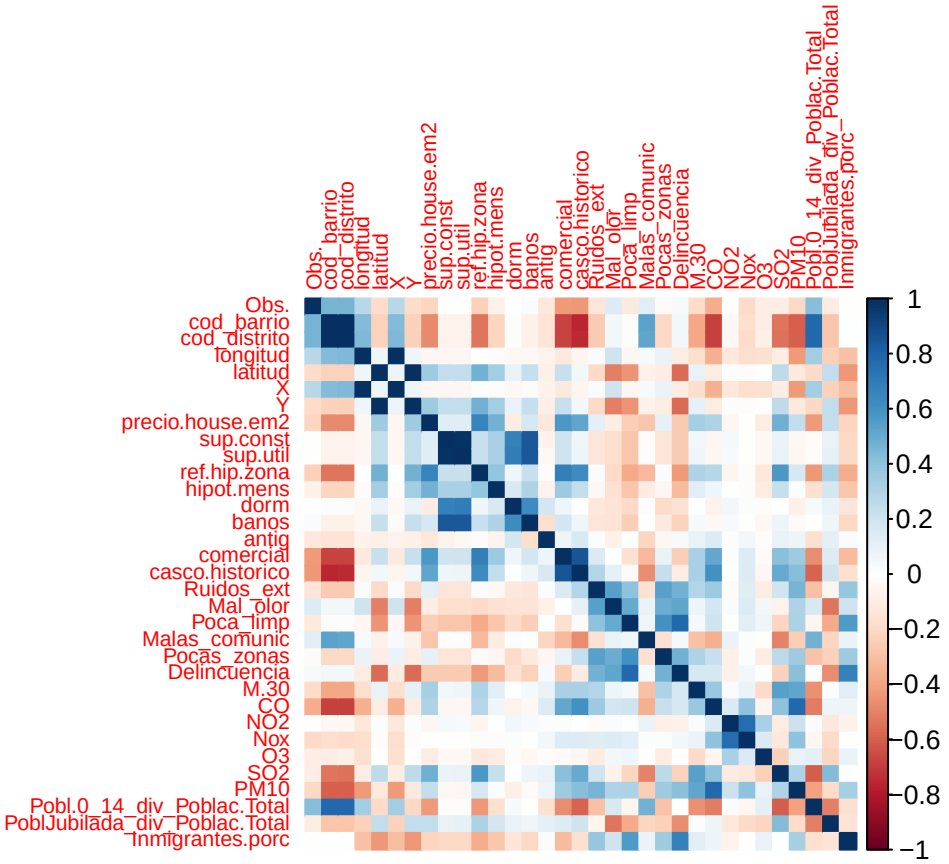
```
##
##      arganzuela         barajas     carabanchel          centro       chamartin
##              69               7              55              85              57
##        chamberi   ciudad_lineal      fuencarral       hortaleza          latina
##              20              68              50              25              61
##         moncloa       moratalaz puente_vallecas         retiro       salamanca
##              52              18              74              42              65
##        san_blas          tetuan           usera        vallecas       vicalvaro
##              32              71              55              31              16
##      villaverde
##              47
```

3.3. Análisis de Correlación

Por último vamos a utilizar matrices de correlación para determinar la relación entre los distintos atributos.

```
# Calcular la matriz de correlación
correlation_matrix <- cor(select_if(data, is.numeric), use = "complete.obs")

# Mostrar la matriz de correlación como un heatmap
corrplot(correlation_matrix, method = "color", tl.cex = 0.7)
```



4. Identificación y validación de Reglas de Negocio

A partir de este análisis inicial y la revisión de los datos del repositorio, se pueden inferir una serie de reglas de negocio tales como:

```
# Propuesta de Reglas de Negocio a implementar en el análisis de calidad de datos:

reglas_negocio <- c(
  "Regla 1. Cuando las viviendas de tipo ático tengan una planta asociada que no sea la primera. Para la
,"Regla 2. Cuando las viviends sean de tipo estudio, deben tener al menos un baño y un dormitorio. Por l
,"Regla 3. Será necesario unificar los valores Sí, sí, si, yes, s, etc. para la existencia de trasteros
,"Regla 4. El valor de los años de antiguedad debes ser entero para que no aparezcan decimales."
)

# Mostrar las reglas de negocio propuestas
reglas_negocio
```

```
## [1] "Regla 1. Cuando las viviendas de tipo ático tengan una planta asociada que no sea la primera. Pa
## [2] "Regla 2. Cuando las viviends sean de tipo estudio, deben tener al menos un baño y un dormitorio
## [3] "Regla 3. Será necesario unificar los valores Sí, sí, si, yes, s, etc. para la existencia de tras
## [4] "Regla 4. El valor de los años de antiguedad debes ser entero para que no aparezcan decimales."
```

Regla 1. Cuando las viviendas de tipo ático tengan una planta asociada que no sea la primera. Para las viviendas que no venga reflejada la planta, se considerará que la planta será la sexta.

```
# Chequeo de viviendas de tipo ático que estén en la planta 'primera' o sin planta informada (NA)
resultado_regla_1 <- data %>%
  filter(tipo.house. == "atico" & (is.na(planta.house.) | planta.house. == "primera"))

# Imprimir las viviendas que cumplan con las condiciones de chequeo
nrow(resultado_regla_1)
```

```
## [1] 14
```

Regla 2. Cuando las viviends sean de tipo estudio, deben tener al menos un baño y un dormitorio. Por lo tanto, habrá que corregir esta casuistica cuando los valores sean de 0.

```
# Chequeo de viviendas de tipo estudio que no tengan al menos un baño o un dormitorio
resultado_regla_2 <- data %>%
  filter(tipo.house. == "estudio" & (is.na(banos) | banos < 1 | is.na(dorm) | dorm < 1))

# Imprimir las viviendas que no cumplan con la regla
nrow(resultado_regla_2)
```

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

Regla 3. Será necesario unificar los valores Sí, sí, si, yes, s, etc. para la existencia de trasteros, de forma que solo pueda ser 'sí', y 'no'. En caso que el valor no venga informado, se considerará como 'no'.

```
# Chequeo de valores inválidos en la columna 'trastero'
resultado_regla_3 <- data %>%
  filter(!(trastero. %in% c("no", "sí")))

# Imprimir las viviendas con valores inválidos en la variable 'trastero'
nrow(resultado_regla_3)
```

```
## [1] 464
```

Regla 4. El valor de los años de antiguedad debes ser entero para que no aparezcan decimales.

```
# Verificar si hay valores decimales en la columna antig
resultado_regla_4 <- data %>% filter(antig %% 1 != 0)
nrow(resultado_regla_4)
```

```
## [1] 277
```

Resumen de Resultados y Conclusiones

```
# Resumen del grado de cumplimiento de las reglas
resumen_reglas <- data.frame(
  Regla = c( "Regla 1. Cuando las viviendas de tipo ático tengan una planta asociada que no sea la prime
             "Regla 2. Cuando las viviends sean de tipo estudio, deben tener al menos un baño y un dormi
             "Regla 3. Será necesario unificar los valores Sí, sí, si, yes, s, etc. para la existencia d
             "Regla 4. El valor de los años de antiguedad debes ser entero para que no aparezcan decimal

  Cumplimiento = c(nrow(resultado_regla_1) == 0,
                   nrow(resultado_regla_2) == 0,
                   nrow(resultado_regla_3) == 0,
                   nrow(resultado_regla_4) == 0)
)

# Mostrar el resumen
resumen_reglas
```

```
##
## 1                      Regla 1. Cuando las viviendas de tipo ático tengan una planta asociada que no
## 2                                  Regla 2. Cuando las viviends sean de tipo estudio, deben tener
## 3 Regla 3. Será necesario unificar los valores Sí, sí, si, yes, s, etc. para la existencia de traste
## 4
##   Cumplimiento
## 1        FALSE
## 2        FALSE
## 3        FALSE
## 4        FALSE
```

Como se puede haber, no se cumplen estas reglas de negocio, por lo que deberiamos solucionar estos incumplimientos para que los datos de vivienda tengn un mejor nivel de calidad y con ello maximizar su posibilidad de uso.

5. Mejora de calidad de datos para las reglas no cumplidas

Regla 1. Cuando las viviendas de tipo ático tengan una planta asociada que no sea la primera. Para las viviendas que no venga reflejada la planta, se considerará que la planta será la sexta.

```
# Paso 1: Estandarización del tipo de vivienda (opcional, si es necesario)
data <- data %>%
  mutate(tipo.house. = tolower(tipo.house.))  # Convierte a minúsculas para estandarizar

# Paso 2: Corrección de registros sin planta (asignar planta 'sexta' si es NA para áticos)
data <- data %>%
  mutate(planta.house. = ifelse(tipo.house. == "atico" & is.na(planta.house.), "sexta", planta.house.))

# Paso 3: Chequeo de viviendas de tipo ático que estén en la planta 'primera'
resultado_regla_1 <- data %>%
  filter(tipo.house. == "atico" & planta.house. == "primera")
```

```r
# Imprimir la cantidad de viviendas que incumplen la regla (ático en planta 'primera')
nrow(resultado_regla_1)
```

```
## [1] 0
```

Regla 2. Cuando las viviends sean de tipo estudio, deben tener al menos un baño y un dormitorio. Por lo tanto, habrá que corregir esta casuistica cuando los valores sean de 0.

```r
# Paso 1: Corrección de baños y dormitorios para estudios con valores 0 (asignar al menos 1)
data <- data %>%
  mutate(
    banos = ifelse(tipo.house. == "estudio" & (is.na(banos) | banos < 1), 1, banos),
    dorm = ifelse(tipo.house. == "estudio" & (is.na(dorm) | dorm < 1), 1, dorm)
  )

# Paso 2: Chequeo de viviendas de tipo estudio que aún no cumplan la regla (caso residual)
resultado_regla_2 <- data %>%
  filter(tipo.house. == "estudio" & (banos < 1 | dorm < 1))

# Imprimir la cantidad de viviendas que no cumplieron la regla (caso excepcional)
nrow(resultado_regla_2)
```

```
## [1] 0
```

Regla 3. Será necesario unificar los valores Sí, sí, si, yes, s, etc. para la existencia de trasteros, de forma que solo pueda ser 'sí', y 'no'. En caso que el valor no venga informado, se considerará como 'no'.

```r
# Paso 1: Estandarización de valores en la columna 'trastero'
data <- data %>%
  mutate(
    trastero. = case_when(
      tolower(trastero.) %in% c("sí", "si", "s", "yes", "y", "sii", "si!") ~ "sí",  # Variaciones que s
      tolower(trastero.) %in% c("no", "n", "na", "") ~ "no",  # Variaciones que significan 'no' o no in
      TRUE ~ "no"  # Valor por defecto si no está informado
    )
  )

# Paso 2: Chequeo de valores aún inválidos en la columna 'trastero' (si queda alguno)
resultado_regla_3 <- data %>%
  filter(!(trastero. %in% c("no", "sí")))

# Imprimir la cantidad de viviendas que aún tienen valores inválidos (caso muy excepcional)
nrow(resultado_regla_3)
```

```
## [1] 0
```

Regla 4. El valor de los años de antiguedad debes ser entero para que no aparezcan decimales.

```
# Verificar si hay valores decimales en la columna 'antig'
antiguedad_decimales <- data %>% filter(antig %% 1 != 0)
cat("Número de registros con antigüedad decimal: ", nrow(antiguedad_decimales), "\n")
```

```
## Número de registros con antigüedad decimal:  277
```

```
# Modificar los valores de la columna 'antig' para que sean enteros (redondear al número más cercano)
data <- data %>% mutate(antig = round(antig, 0))

# Verificar si los valores se han corregido
antiguedad_decimales_post <- data %>% filter(antig %% 1 != 0)
cat("Número de registros con antigüedad decimal después de la corrección: ", nrow(antiguedad_decimales_
```

```
## Número de registros con antigüedad decimal después de la corrección:  0
```

Tarea a realizar

A continuación se proponen 3 ejercicios a completar. Se recomienda completaros en script markdown de R que se proporciona.

Tarea 1

Hacer un análisis exploratorio de los datos similar al realizado en este ejercicio sobre la conexión a la base de datos que ya tenéis del módulo de bases de datos relacionales

```
print("Completar Tarea 1")
```

```
## [1] "Completar Tarea 1"
```

Tarea 2

Una vez realizado el análisis exploratorio, crear los scripts para validar el grado de cumplimiento de estas reglas de negocio:

Cuando la antigüedad de la vivienda sea de 0 años, el estado no podrá ser a reformar, si no que se corregira por defeco a 'buen_estado'

Será necesario unificar los valores Sí, sí, si, yes, s, etc. para la existencia de ascensor, de forma que solo pueda ser 'sí', y 'no'. En caso que el valor no venga informado, se considerará como 'no'.

Se considerará que todo lo que esté en la zona interior de la M.30 será considerado como vivienda de tipo zona comercial, por lo que habrá que corregir el valor para indicar que se encuentra en zona comercial.

```
print("Completar Tarea 2")
```

```
## [1] "Completar Tarea 2"
```

Tarea 3

Corregir los problemas de calidad de datos identificados para poder cumplir con las reglas de negocio y llegar al 100% de grado de cumplimiento

```
print("Completar Tarea 3")
```

```
## [1] "Completar Tarea 3"
```