

Almacenes y Minería de Datos



PROYECTO

SISTEMA DE MINERÍA DE DATOS

Carlos Cruz Rangel

Miguel Angel Liera Montaña

Alejandro Urbano Flores

ÍNDICE

Sistema de Minería de Datos	3
1. Descripción del problema	3
2. Conocimiento de los datos (análisis exploratorio)	3
Ejercicio - 2a -	5
TABLA:	15
Ejercicio - 2b -	25
Ejercicio - 2c -	28
Ejercicio - 2d -	34
Ejercicio - 2e -	34
3. Preprocesamiento de datos	36
Selección de atributos.	36
Manejo de valores perdidos.	37
Eliminación de valores atípicos.	39
Discretización de atributos numéricos.	41
Normalización.	43
4. Minado de datos	45
Utiliza un árbol CART.	45
La salida del árbol modelado fue el siguiente:	46
Red Neuronal	47
Evaluación de modelos de clasificación.	49
Utiliza reglas de asociación.	50
Utiliza agrupación.	53
5. Conclusiones	53

1. Descripción del problema

El objetivo de este proyecto es desarrollar un sistema de minería de datos, utilizando la metodología CRISP. El problema consiste en que de acuerdo con los resultados obtenidos se propondrán algunas medidas y/o mejores prácticas sobre algunos de los conjuntos de datos que se indican en la presente especificación.

2. Conocimiento de los datos (análisis exploratorio)

Airbnb ha experimentado un crecimiento meteórico desde su creación en 2008, y la cantidad de alquileres que figuran en su sitio web crece exponencialmente cada año. Esta plataforma digital ha revolucionado con éxito la industria hotelera tradicional a medida que más y más viajeros la utilizan, no solo los que buscan sacar partido a su inversión, sino también los viajeros de negocios que recurren a Airbnb como su principal proveedor de alojamiento.

LINK PARA DATA SET

En este proyecto se trabajará con un dataset que posee un conjunto de datos recopilados, principalmente de Inside Airbnb. Algunas de las columnas del dataset son:

- **room_id:** un número único que identifica un listado de Airbnb. El listado tiene una URL en el sitio web de Airbnb de http://airbnb.com/rooms/room_id.
- **host_id:** un número único que identifica a un anfitrión de Airbnb. La página del anfitrión tiene una URL en el sitio web de Airbnb de http://airbnb.com/users/show/host_id.
- **room_type:** alguno de los siguientes valores: "Entire home/apt", "Private room", or "Shared room".

- **borough:** Una subregión de la ciudad o área de búsqueda para la cual se lleva a cabo la encuesta. Para algunas ciudades, no hay información del municipio; para otros, el municipio puede ser un número.
- **neighborhood:** Al igual que municipio: una subregión de la ciudad o área de búsqueda para la que se realiza la encuesta. Para las ciudades que tienen ambos, un vecindario es más pequeño que un municipio. Para algunas ciudades no hay información del vecindario.
- **Numero_of_reviews:** el número de reseñas que ha recibido un anuncio. Airbnb ha dicho que el 70% de las visitas terminan con una reseña, por lo que la cantidad de reseñas se puede usar para estimar la cantidad de visitas. Tenga en cuenta que dicha estimación no será confiable para una lista individual (especialmente porque las revisiones ocasionalmente desaparecen del sitio), pero en una ciudad en su conjunto debería ser una métrica útil de tráfico.
- **review_scores_rating:** La calificación promedio que la lista ha recibido de aquellos visitantes que dejaron una reseña
- **accommodates:** el número de invitados que puede acomodar un anuncio.
- **bedrooms:** el número de dormitorios que ofrece un anuncio.
- **price:** El precio (en \$US) por una noche de estadía.
- **minimum_nights/maximum_nights:** La estadía mínima/máxima para una visita, según lo publicado por el anfitrión.
- **latitude/longitude:** la latitud y la longitud del listado tal como se publicó en el sitio de Airbnb: esto puede estar desviado por unos cientos de metros.
- **last_modified:** la fecha y la hora en que se leyeron los valores del sitio web de Airbnb

Ejercicio - 2a -

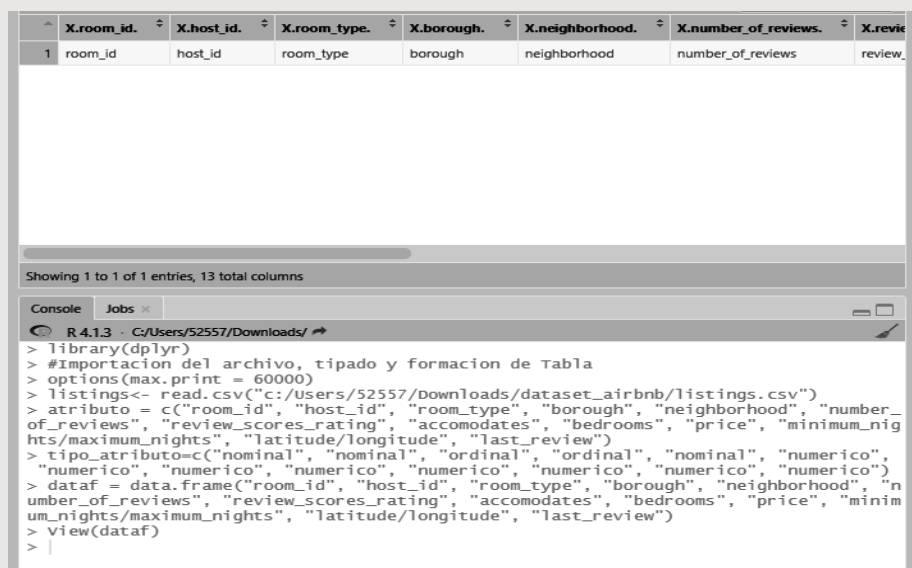
Elaboración de una tabla con la siguiente información para cada atributo:

- I. Tipo de atributo (nominal, ordinal, numérico, etc.).

Para esta parte se tomaron los atributos antes dados en el dataset y agregamos otros más, que se consideran importantes para el análisis de los datos; dichos atributos agregados son:

- ☐ Guests_Included
- ☐ Bedrooms
- ☐ Bathrooms
- ☐ Square_Feet
- ☐ Bed_type
- ☐ Room_type
- ☐ Property_type

A continuación anexamos la captura de pantalla con los atributos:



```
> library(dplyr)
> #Importacion del archivo, tipado y formacion de Tabla
> options(max.print = 60000)
> listings<- read.csv("c:/Users/52557/Downloads/dataset_airbnb/listings.csv")
> atributo = c("room_id", "host_id", "room_type", "borough", "neighborhood", "number_of_reviews", "review_scores_rating", "accommodates", "bedrooms", "price", "minimum_nights", "maximum_nights", "latitude/longitude", "last_review")
> tipo_atributo=c("nominal", "nominal", "ordinal", "ordinal", "nominal", "numerico", "numerico", "numerico", "numerico", "numerico", "numerico", "numerico", "numerico", "numerico")
> dataf = data.frame("room_id", "host_id", "room_type", "borough", "neighborhood", "number_of_reviews", "review_scores_rating", "accommodates", "bedrooms", "price", "minimum_nights", "maximum_nights", "latitude/longitude", "last_review")
> view(dataf)
> |
```


II. Porcentaje de valores perdidos.

Para el porcentaje de valores, se observaron la o las columnas del dataset que tuvieran valores **na** (valores faltantes o perdidos).

Los porcentajes de valores perdidos por atributos son:

- ☐ Review_Scores_Rating: 22.98%
- ☐ Last_Review: 20.84%
- ☐ Square_Feet: 99.03%
- ☐ Bed_Rooms: 0.10%
- ☐ Bathrooms: 0.19%

Anexamos captura de pantalla del código implementado para los datos perdidos:

```
> #.....  
> #Valores perdidos  
> #.....  
> #is.na.data.frame(listings$review_scores_rating)  
> sum(is.na.data.frame(listings$review_scores_rating))  
[1] 11714  
> sum(complete.cases(data.frame(listings$review_scores_rating)))  
[1] 39254  
> #Las únicas variables con datos perdidos son las siguientes, cuyo porcentaje es:  
> mean(is.na(listings$review_scores_rating))  
[1] 0.2298305  
> mean(is.na(listings$last_review))  
[1] 0.2084641  
> mean(is.na(listings$square_feet))  
[1] 0.9903273  
> mean(is.na(listings$bedrooms))  
[1] 0.001098729  
> mean(is.na(listings$bathrooms))  
[1] 0.001922775  
> #Las demás no cuentan con datos perdidos:  
> mean(is.na(listings$number_of_reviews))  
[1] 0  
> mean(is.na(listings$accommodates))  
[1] 0  
> mean(is.na(listings$price))  
[1] 0  
> mean(is.na(listings$minimum_nights))  
[1] 0  
> mean(is.na(listings$maximum_nights))  
[1] 0  
> mean(is.na(listings$latitude))  
[1] 0  
> mean(is.na(listings$longitude))  
[1] 0  
> mean(is.na(listings$guests_included))  
[1] 0  
> mean(is.na(listings$bed_type))  
[1] 0  
> mean(is.na(listings$room_type))  
[1] 0  
.. .. .. .. ..
```

Pudimos ver que sólo los siguientes atributos no contaron con valores perdidos:

- ☐ Number_of_reviews
- ☐ Accommodates
- ☐ Price
- ☐ Minimum_Nights
- ☐ Maximum_Nights
- ☐ Latitude
- ☐ Longitude
- ☐ Guests_included
- ☐ Bed_type
- ☐ Rooms_type
- ☐ Property_type

III. Valor mínimo, máximo, media, desviación estándar (si aplica).

A continuación veremos los valores mínimos con su respectivo código:

```
> #.....  
> #valores minimos  
> #.....  
> min(listings$number_of_reviews)  
[1] 0  
> min(na.omit(listings$review_scores_rating))  
[1] 20  
> min(listings$accommodates)  
[1] 1  
> min(listings$price)  
[1] 0  
> min(listings$minimum_nights)  
[1] 1  
> min(listings$maximum_nights)  
[1] 1  
> min(listings$latitude)  
[1] 40.49979  
> min(listings$longitude)  
[1] -74.24084  
> min(na.omit(listings$last_review))  
[1] 40344  
> min(listings$guests_included)  
[1] 1  
> min(na.omit(listings$bedrooms))  
[1] 0  
> min(na.omit(listings$bathrooms))  
[1] 0  
> min(na.omit(listings$square_feet))  
[1] 0  
> |
```

A continuación mostramos los valores máximos y su respectivo código:

```
> #.....  
> # valores maximos  
> #.....  
> max(listings$number_of_reviews)  
[1] 557  
> max(na.omit(listings$review_scores_rating))  
[1] 100  
> max(listings$accommodates)  
[1] 16  
> max(listings$price)  
[1] 10000  
> max(listings$minimum_nights)  
[1] 1250  
> max(listings$maximum_nights)  
[1] 2147483647  
> max(listings$latitude)  
[1] 40.91171  
> max(listings$longitude)  
[1] -73.708  
> max(na.omit(listings$last_review))  
[1] 43407  
> max(listings$guests_included)  
[1] 16  
> max(na.omit(listings$bedrooms))  
[1] 14  
> max(na.omit(listings$bathrooms))  
[1] 16.5  
> max(na.omit(listings$square_feet))  
[1] 5000  
> |
```


En esta parte mostramos la media de los valores y su código:

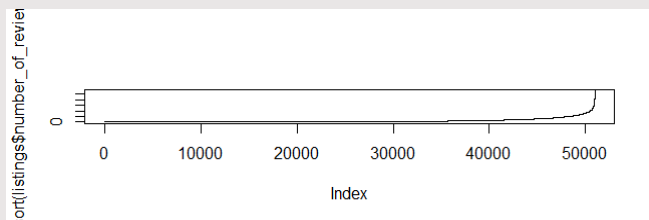
```
> #.....
> # Media
> #.....
> mean(na.omit(listings$review_scores_rating))
[1] 93.6825
> mean(listings$number_of_reviews)
[1] 21.48499
> mean(listings$accommodates)
[1] 2.862168
> mean(listings$minimum_nights)
[1] 6.843255
> mean(listings$maximum_nights)
[1] 43585.85
> mean(listings$latitude)
[1] 40.73003
> mean(listings$longitude)
[1] -73.95366
> mean(listings$price)
[1] 151.2304
> mean(listings$guests_included)
[1] 1.503394
> mean(na.omit(listings$bedrooms))
[1] 1.176972
> mean(na.omit(listings$bathrooms))
[1] 1.142766
> mean(na.omit(listings$square_feet))
[1] 720.501
> mean(na.omit(listings$last_review))
[1] 43178.34
```

Aquí podemos ver el código implementado para la desviación estándar:

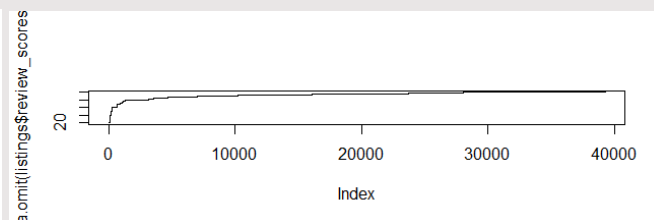
```
> #.....
> # Desviacion Estandar
> #.....
> sd(listings$number_of_reviews)
[1] 40.1676
> sd(na.omit(listings$review_scores_rating))
[1] 8.510003
> sd(listings$accommodates)
[1] 1.880318
> sd(listings$price)
[1] 221.8948
> sd(listings$minimum_nights)
[1] 19.58203
> sd(listings$maximum_nights)
[1] 9513023
> sd(listings$latitude)
[1] 0.0537904
> sd(listings$longitude)
[1] 0.04417501
> sd(na.omit(listings$last_review))
[1] 350.6567
> sd(listings$guests_included)
[1] 1.128416
> sd(na.omit(listings$bedrooms))
[1] 0.7526933
> sd(na.omit(listings$bathrooms))
[1] 0.4338172
> sd(na.omit(listings$square_feet))
[1] 584.6176
```

IV. Si es numérico, indicar el tipo de distribución que parece seguir (p.e. normal).

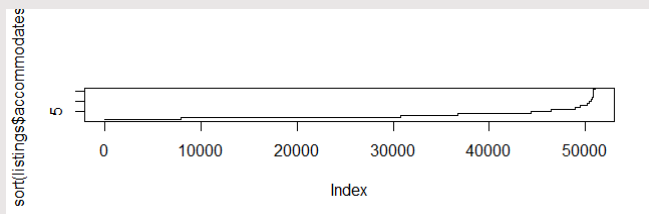
Mostraremos el tipo de distribución de los atributos:



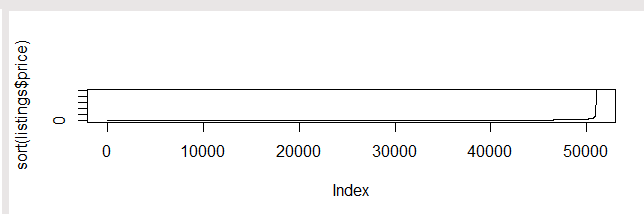
Number_Of_Reviews



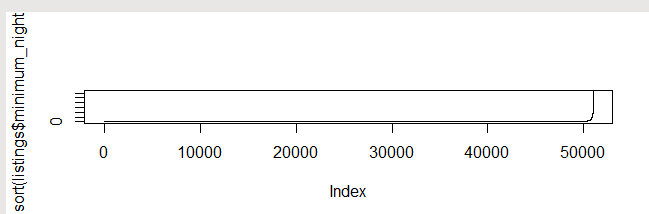
Review_Scores_Rating



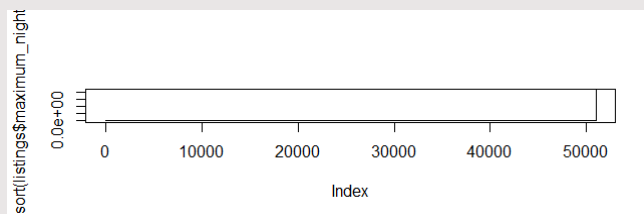
Accomodates



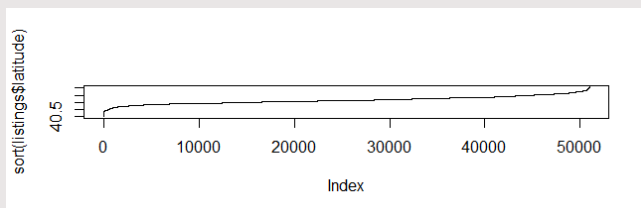
Price



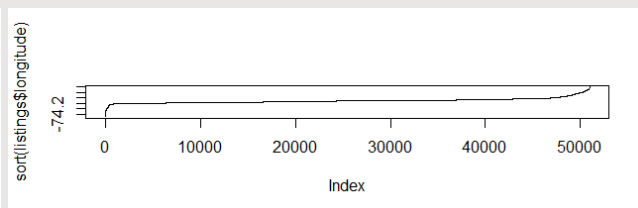
Minimum_Nights



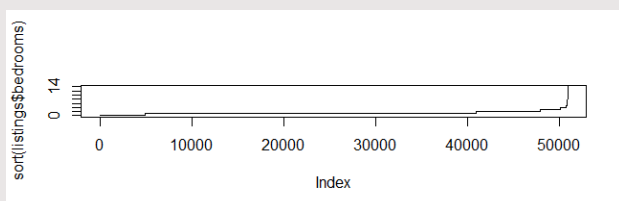
Maximum_Nights



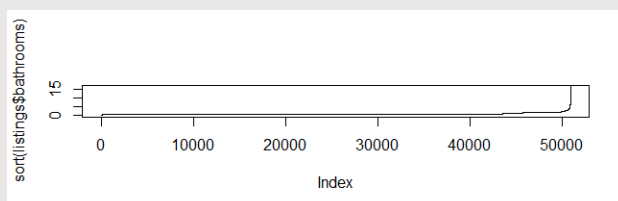
Latitude



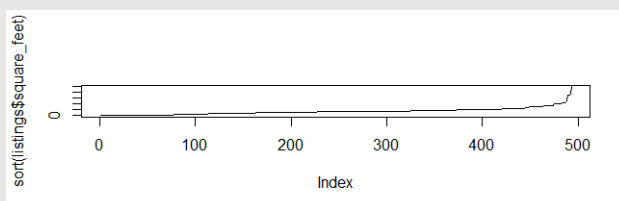
Longitude



Bedrooms



Bathrooms



Square_Feet

Vimos los tipos de distribución de algunos atributos.

En la tabla que se nos pedía en un inicio marcamos los tipos de distribución a los que se asemejan, pero eso lo veremos poco más adelante.

- V. Indicar si existen registros que tengan un valor para ese atributo, que no aparezca en otros registros.

Nosotros notamos que los valores para cada atributo eran únicos y por tanto no aparecían en ningún otro atributo o registro.

VI. Indicar si el atributo presenta valores atípicos

Para los valores atípicos mostramos la gráfica de caja de cada atributo y observamos lo siguiente:

```
> #valores atipicos
> #Para los valores atipicos mostramos la gráfica de caja de la variable y observamos esto
>
> #number_of_reviews si tiene datos atipicos
> boxplot(listings$number_of_reviews)
> #review_scores_rating si tiene atipicos
> boxplot(na.omit(listings$review_scores_rating))
> #accommodates si tiene atipicos
> boxplot(listings$accommodates)
> #price si tiene atipicos
> boxplot(listings$price)
> #minimum_nights si tiene atipicos
> boxplot(listings$minimum_nights)
> #maximum_nights si tiene atipicos
> boxplot(na.omit(listings$maximum_nights))

> #latitude_nights si tiene atipicos
> boxplot(listings$latitude)
> #longitude si tiene atipicos
> boxplot(listings$longitude)
> #last_review si tiene atipicos
> boxplot(na.omit(listings$last_review))
> #guests_included si tiene atipicos
> boxplot(listings$guests_included)
> #bedrooms si tiene atipicos
> boxplot(na.omit(listings$bedrooms))
> #bedrooms si tiene atipicos
> boxplot(na.omit(listings$bedrooms))
> #square_feet si tiene atipicos
> boxplot(na.omit(listings$square_feet))
>
> #Los siguientes datos no tiene sentido eliminar datos atipicos
> #guests_included

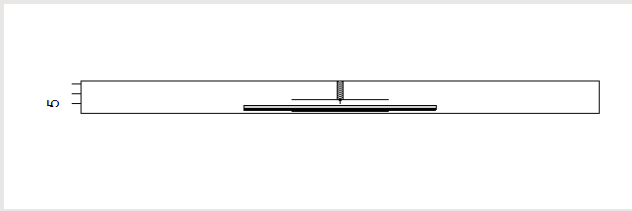
> #bedrooms
> #bathrooms
```



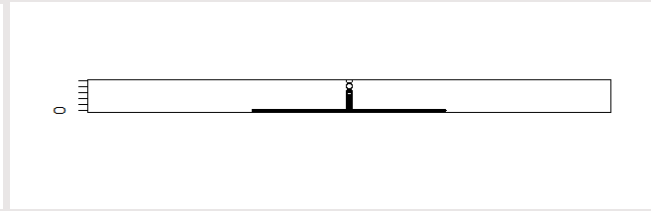
Number_Of_Reviews



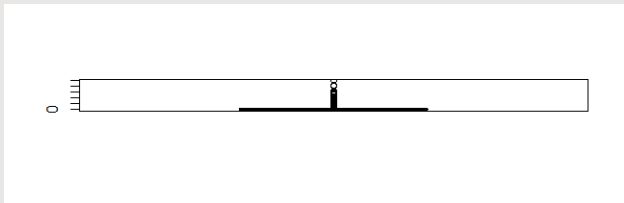
Review_Scores_Rating



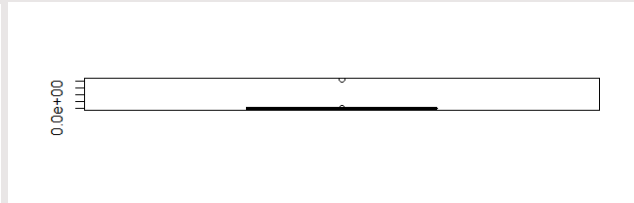
Accomates



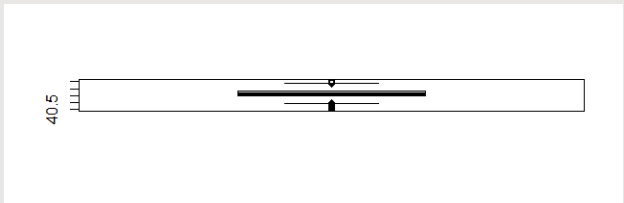
Price



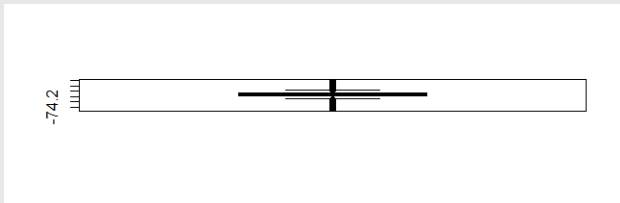
Minimum_Nights



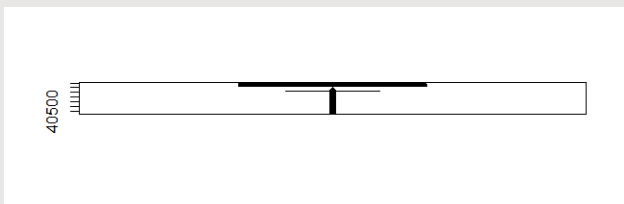
Maximum_Nights



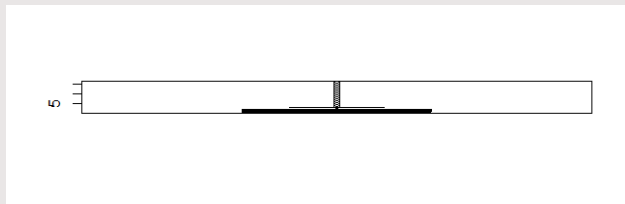
Latitude



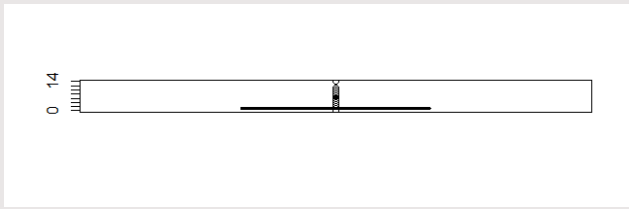
Longitude



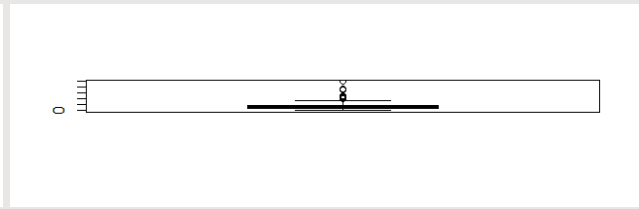
Last_Review



Guests_Included



Bedrooms



Square_Feet

TABLA:

En la siguiente tabla presentamos todos los datos antes mostrados.

ATRIBUTO	VALORES		VALORES ATÍPICOS
Room_id	Tipo	Nominal	No presentó valores atípicos.
	% Valores Perdidos	0%	
	Máximo	no aplica	
	Mínimo	no aplica	
	Media	no aplica	
	Desviación Estándar	no aplica	
	Distribución	no aplica	
Host_id	Tipo	Nominal	No presentó valores atípicos.
	% Valores Perdidos	0%	
	Máximo	no aplica	
	Mínimo	no aplica	
	Media	no aplica	
	Desviación Estándar	no aplica	
	Distribución	no aplica	

	<table><tr><td>n</td><td></td></tr></table>	n														
n																
Borough	<table><tr><td>Tipo</td><td>Ordinal</td></tr><tr><td>% Valores Perdidos</td><td>0%</td></tr><tr><td>Máximo</td><td>no aplica</td></tr><tr><td>Mínimo</td><td>no aplica</td></tr><tr><td>Media</td><td>no aplica</td></tr><tr><td>Desviación Estándar</td><td>no aplica</td></tr><tr><td>Distribución</td><td>no aplica</td></tr></table>	Tipo	Ordinal	% Valores Perdidos	0%	Máximo	no aplica	Mínimo	no aplica	Media	no aplica	Desviación Estándar	no aplica	Distribución	no aplica	
Tipo	Ordinal															
% Valores Perdidos	0%															
Máximo	no aplica															
Mínimo	no aplica															
Media	no aplica															
Desviación Estándar	no aplica															
Distribución	no aplica															
Neighborhood	<table><tr><td>Tipo</td><td>Nominal</td></tr><tr><td>% Valores Perdidos</td><td>0%</td></tr><tr><td>Máximo</td><td>no aplica</td></tr><tr><td>Mínimo</td><td>no aplica</td></tr><tr><td>Media</td><td>no aplica</td></tr><tr><td>Desviación Estándar</td><td>no aplica</td></tr><tr><td>Distribución</td><td>no aplica</td></tr></table>	Tipo	Nominal	% Valores Perdidos	0%	Máximo	no aplica	Mínimo	no aplica	Media	no aplica	Desviación Estándar	no aplica	Distribución	no aplica	No presentó valores atípicos.
Tipo	Nominal															
% Valores Perdidos	0%															
Máximo	no aplica															
Mínimo	no aplica															
Media	no aplica															
Desviación Estándar	no aplica															
Distribución	no aplica															
Number_Of_Review		Tiene valores atípicos														

	<table><tr><td>Tipo</td><td>Integer</td></tr><tr><td>% Valores Perdidos</td><td>0%</td></tr><tr><td>Máximo</td><td>557</td></tr><tr><td>Mínimo</td><td>0</td></tr><tr><td>Media</td><td>21.48499</td></tr><tr><td>Desviación Estándar</td><td>40.1676</td></tr><tr><td>Distribución</td><td>Exponencial</td></tr></table>	Tipo	Integer	% Valores Perdidos	0%	Máximo	557	Mínimo	0	Media	21.48499	Desviación Estándar	40.1676	Distribución	Exponencial	
Tipo	Integer															
% Valores Perdidos	0%															
Máximo	557															
Mínimo	0															
Media	21.48499															
Desviación Estándar	40.1676															
Distribución	Exponencial															
Review_Scores_Rating	<table><tr><td>Tipo</td><td>Integer</td></tr><tr><td>% Valores Perdidos</td><td>22.98%</td></tr><tr><td>Máximo</td><td>100</td></tr><tr><td>Mínimo</td><td>20</td></tr><tr><td>Media</td><td>93.6825</td></tr><tr><td>Desviación Estándar</td><td>8.510003</td></tr><tr><td>Distribución</td><td>Log Normal</td></tr></table>	Tipo	Integer	% Valores Perdidos	22.98%	Máximo	100	Mínimo	20	Media	93.6825	Desviación Estándar	8.510003	Distribución	Log Normal	Tiene valores atípicos
Tipo	Integer															
% Valores Perdidos	22.98%															
Máximo	100															
Mínimo	20															
Media	93.6825															
Desviación Estándar	8.510003															
Distribución	Log Normal															
Accommodates	<table><tr><td>Tipo</td><td>Integer</td></tr><tr><td>% Valores Perdidos</td><td>0%</td></tr></table>	Tipo	Integer	% Valores Perdidos	0%	Tiene valores atípicos										
Tipo	Integer															
% Valores Perdidos	0%															

	<table><tr><td>Máximo</td><td>16</td></tr><tr><td>Mínimo</td><td>1</td></tr><tr><td>Media</td><td>2.862168</td></tr><tr><td>Desviación Estándar</td><td>1.880318</td></tr><tr><td>Distribución</td><td>Exponencial</td></tr><tr><td colspan="2"></td></tr></table>	Máximo	16	Mínimo	1	Media	2.862168	Desviación Estándar	1.880318	Distribución	Exponencial							
Máximo	16																	
Mínimo	1																	
Media	2.862168																	
Desviación Estándar	1.880318																	
Distribución	Exponencial																	
Price	<table><tr><td>Tipo</td><td>Integer</td></tr><tr><td>% Valores Perdidos</td><td>0%</td></tr><tr><td>Máximo</td><td>10,000</td></tr><tr><td>Mínimo</td><td>0</td></tr><tr><td>Media</td><td>151.2304</td></tr><tr><td>Desviación Estándar</td><td>221.8948</td></tr><tr><td>Distribución</td><td>Exponencial</td></tr><tr><td colspan="2"></td></tr></table>	Tipo	Integer	% Valores Perdidos	0%	Máximo	10,000	Mínimo	0	Media	151.2304	Desviación Estándar	221.8948	Distribución	Exponencial			Tiene valores atípicos
Tipo	Integer																	
% Valores Perdidos	0%																	
Máximo	10,000																	
Mínimo	0																	
Media	151.2304																	
Desviación Estándar	221.8948																	
Distribución	Exponencial																	
Minimum_Nights	<table><tr><td>Tipo</td><td>Integer</td></tr><tr><td>% Valores Perdidos</td><td>0%</td></tr><tr><td>Máximo</td><td>1,250</td></tr><tr><td>Mínimo</td><td>1</td></tr><tr><td>Media</td><td>6.843255</td></tr></table>	Tipo	Integer	% Valores Perdidos	0%	Máximo	1,250	Mínimo	1	Media	6.843255	Tiene valores atípicos						
Tipo	Integer																	
% Valores Perdidos	0%																	
Máximo	1,250																	
Mínimo	1																	
Media	6.843255																	

	<table><tr><td>Desviación Estándar</td><td>19.58203</td></tr><tr><td>Distribución</td><td>Exponencial</td></tr></table>	Desviación Estándar	19.58203	Distribución	Exponencial											
Desviación Estándar	19.58203															
Distribución	Exponencial															
Maximum_Nights	<table><tr><td>Tipo</td><td>Integer</td></tr><tr><td>% Valores Perdidos</td><td>0%</td></tr><tr><td>Máximo</td><td>2,147,483,647</td></tr><tr><td>Mínimo</td><td>1</td></tr><tr><td>Media</td><td>43,585.85</td></tr><tr><td>Desviación Estándar</td><td>9,513,023</td></tr><tr><td>Distribución</td><td>Exponencial</td></tr></table>	Tipo	Integer	% Valores Perdidos	0%	Máximo	2,147,483,647	Mínimo	1	Media	43,585.85	Desviación Estándar	9,513,023	Distribución	Exponencial	Tiene valores atípicos
Tipo	Integer															
% Valores Perdidos	0%															
Máximo	2,147,483,647															
Mínimo	1															
Media	43,585.85															
Desviación Estándar	9,513,023															
Distribución	Exponencial															
Last_Review	<table><tr><td>Tipo</td><td>Integer</td></tr><tr><td>% Valores Perdidos</td><td>20.84%</td></tr><tr><td>Máximo</td><td>43,407</td></tr><tr><td>Mínimo</td><td>40344</td></tr><tr><td>Media</td><td>43585.85</td></tr><tr><td>Desviación Estándar</td><td>350.6567</td></tr></table>	Tipo	Integer	% Valores Perdidos	20.84%	Máximo	43,407	Mínimo	40344	Media	43585.85	Desviación Estándar	350.6567	Tiene valores atípicos		
Tipo	Integer															
% Valores Perdidos	20.84%															
Máximo	43,407															
Mínimo	40344															
Media	43585.85															
Desviación Estándar	350.6567															

	<table><tr><td>Distribución</td><td>No aplica</td></tr></table>	Distribución	No aplica													
Distribución	No aplica															
Guests_Included	<table><tr><td>Tipo</td><td>Integer</td></tr><tr><td>% Valores Perdidos</td><td>0%</td></tr><tr><td>Máximo</td><td>16</td></tr><tr><td>Mínimo</td><td>1</td></tr><tr><td>Media</td><td>1.503394</td></tr><tr><td>Desviación Estándar</td><td>1.128416</td></tr><tr><td>Distribución</td><td>No aplica</td></tr></table>	Tipo	Integer	% Valores Perdidos	0%	Máximo	16	Mínimo	1	Media	1.503394	Desviación Estándar	1.128416	Distribución	No aplica	Tiene valores atípicos
Tipo	Integer															
% Valores Perdidos	0%															
Máximo	16															
Mínimo	1															
Media	1.503394															
Desviación Estándar	1.128416															
Distribución	No aplica															
Bedrooms	<table><tr><td>Tipo</td><td>Integer</td></tr><tr><td>% Valores Perdidos</td><td>0.1%</td></tr><tr><td>Máximo</td><td>14</td></tr><tr><td>Mínimo</td><td>0</td></tr><tr><td>Media</td><td>1.176972</td></tr><tr><td>Desviación Estándar</td><td>0.7526933</td></tr><tr><td>Distribución</td><td>Exponencial</td></tr></table>	Tipo	Integer	% Valores Perdidos	0.1%	Máximo	14	Mínimo	0	Media	1.176972	Desviación Estándar	0.7526933	Distribución	Exponencial	Tiene valores atípicos
Tipo	Integer															
% Valores Perdidos	0.1%															
Máximo	14															
Mínimo	0															
Media	1.176972															
Desviación Estándar	0.7526933															
Distribución	Exponencial															

Square_Feet			Tiene valores atípicos
	Tipo	Integer	
	% Valores perdidos	99.03%	
	Máximo	5,000	
	Mínimo	0	
	Media	720.501	
	Desviación estándar	584.6176	
	Distribución	Exponencial	
	Tipo	Numérico	
	% Valores Perdidos	0%	
	Máximo	40.91171	
	Mínimo	40.49979	
	Media	40.73003	
Latitude	Desviación Estándar	0.0537904	Tiene valores atípicos
	Distribución	Sigmoidal	
Tipo		Numérico	
% Valores Perdidos		0%	

Longitude	Máximo	-73.708	
	Mínimo	-74.24084	
	Media	-73.95366	
	Desviación Estándar	0.04417501	
	Distribución	Sigmoidal	
Bathrooms	Tipo	Numérico	No presentó valores atípicos.
	% Valores Perdidos	0.19%	
	Máximo	16.5	
	Mínimo	0	
	Media	1.142766	
	Desviación Estándar	0.4338172	
	Distribución	Exponencial	
Bed_type	Tipo	Character	No presentó valores atípicos.
	%Valores Perdidos	0%	
	Máximo	No aplica	
	Mínimo	No aplica	

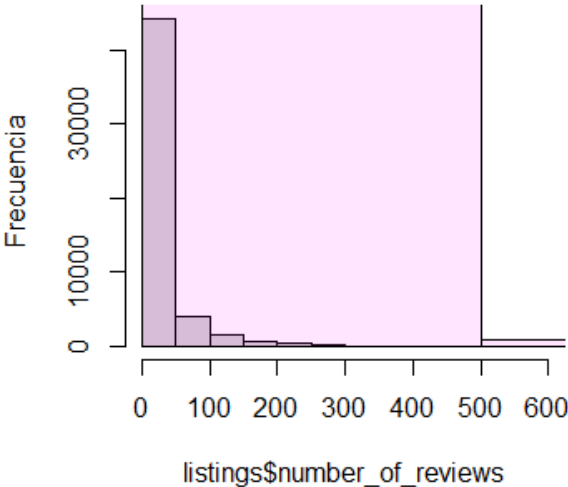
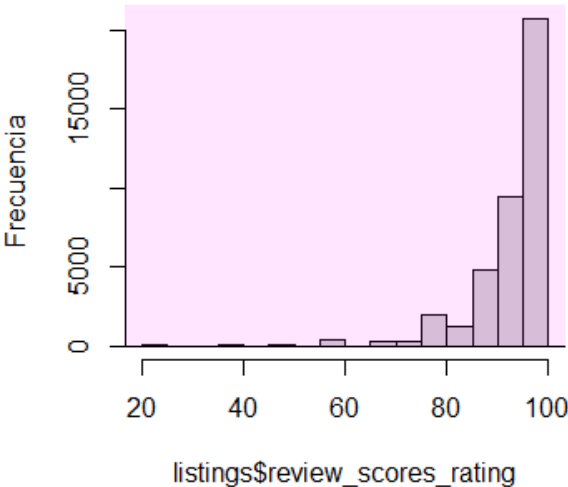
	<table><tr><td>Media</td><td>No aplica</td></tr><tr><td>Desviación Estándar</td><td>No aplica</td></tr><tr><td>Distribución</td><td>No aplica</td></tr></table>	Media	No aplica	Desviación Estándar	No aplica	Distribución	No aplica									
Media	No aplica															
Desviación Estándar	No aplica															
Distribución	No aplica															
Room_type	<table><tr><td>Tipo</td><td>Character</td></tr><tr><td>%Valores Perdidos</td><td>0%</td></tr><tr><td>Máximo</td><td>No aplica</td></tr><tr><td>Mínimo</td><td>No aplica</td></tr><tr><td>Media</td><td>No aplica</td></tr><tr><td>Desviación Estándar</td><td>No aplica</td></tr><tr><td>Distribución</td><td>No aplica</td></tr></table>	Tipo	Character	%Valores Perdidos	0%	Máximo	No aplica	Mínimo	No aplica	Media	No aplica	Desviación Estándar	No aplica	Distribución	No aplica	No presentó valores atípicos.
Tipo	Character															
%Valores Perdidos	0%															
Máximo	No aplica															
Mínimo	No aplica															
Media	No aplica															
Desviación Estándar	No aplica															
Distribución	No aplica															
Property_type	<table><tr><td>Tipo</td><td>Character</td></tr><tr><td>%Valores Perdidos</td><td>0%</td></tr><tr><td>Máximo</td><td>No aplica</td></tr><tr><td>Mínimo</td><td>No aplica</td></tr><tr><td>Media</td><td>No aplica</td></tr><tr><td>Desviación Estándar</td><td>No aplica</td></tr></table>	Tipo	Character	%Valores Perdidos	0%	Máximo	No aplica	Mínimo	No aplica	Media	No aplica	Desviación Estándar	No aplica	No presentó valores atípicos.		
Tipo	Character															
%Valores Perdidos	0%															
Máximo	No aplica															
Mínimo	No aplica															
Media	No aplica															
Desviación Estándar	No aplica															

	Distribució n	No aplica	

Ejercicio - 2b -

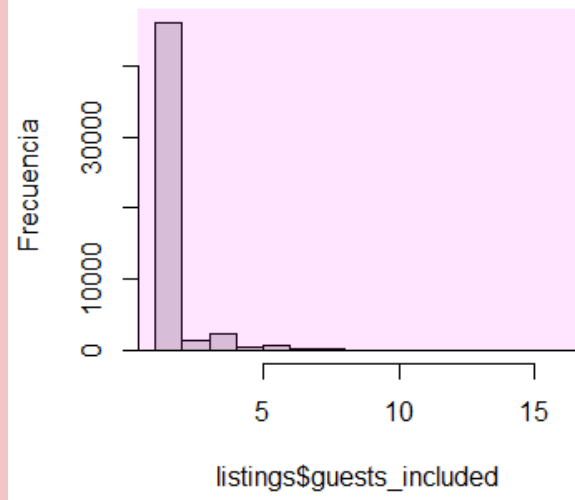
Elaboración de un histograma para determinar cuáles atributos ejercen influencia de acuerdo con la variable objetivo que se definió.

Variable objetivo : **price**

Atributo ligado con el objetivo	<u>Histograma</u>
number_of_reviews	<p data-bbox="1019 619 1333 655">variable Y y objetivo</p>  <p>The histogram shows the frequency distribution of the number of reviews. The x-axis is labeled 'listings\$number_of_reviews' and ranges from 0 to 600. The y-axis is labeled 'Frecuencia' and ranges from 0 to 30,000. The distribution is highly right-skewed, with a very high frequency (over 30,000) for the first bin (0-50 reviews) and a rapid decline in frequency as the number of reviews increases.</p>
review_scores_rating	<p data-bbox="1019 1306 1333 1341">variable Y y objetivo</p>  <p>The histogram shows the frequency distribution of the review scores rating. The x-axis is labeled 'listings\$review_scores_rating' and ranges from 20 to 100. The y-axis is labeled 'Frecuencia' and ranges from 0 to 15,000. The distribution is right-skewed, with most listings having a rating between 80 and 100, and a peak frequency of over 15,000 for the highest rating bin (90-100).</p>

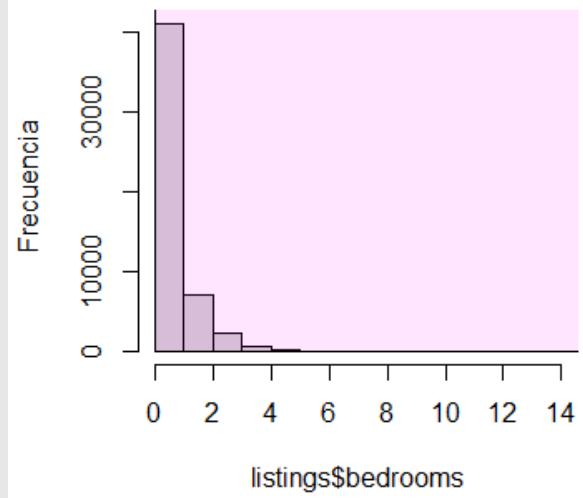
guest_included

variable Y y objetivo

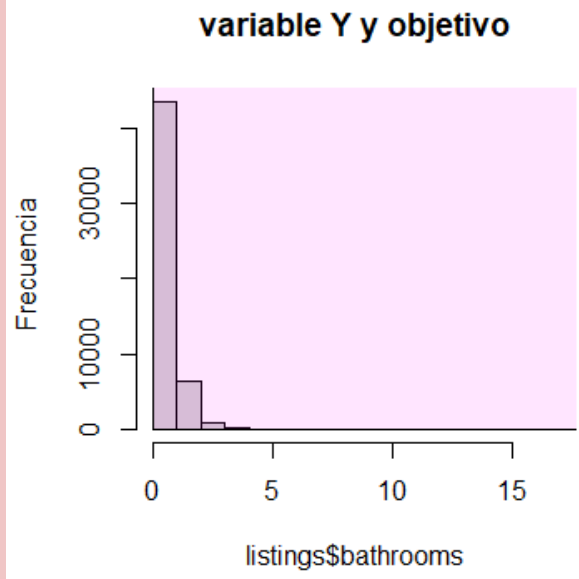


bedrooms

variable Y y objetivo

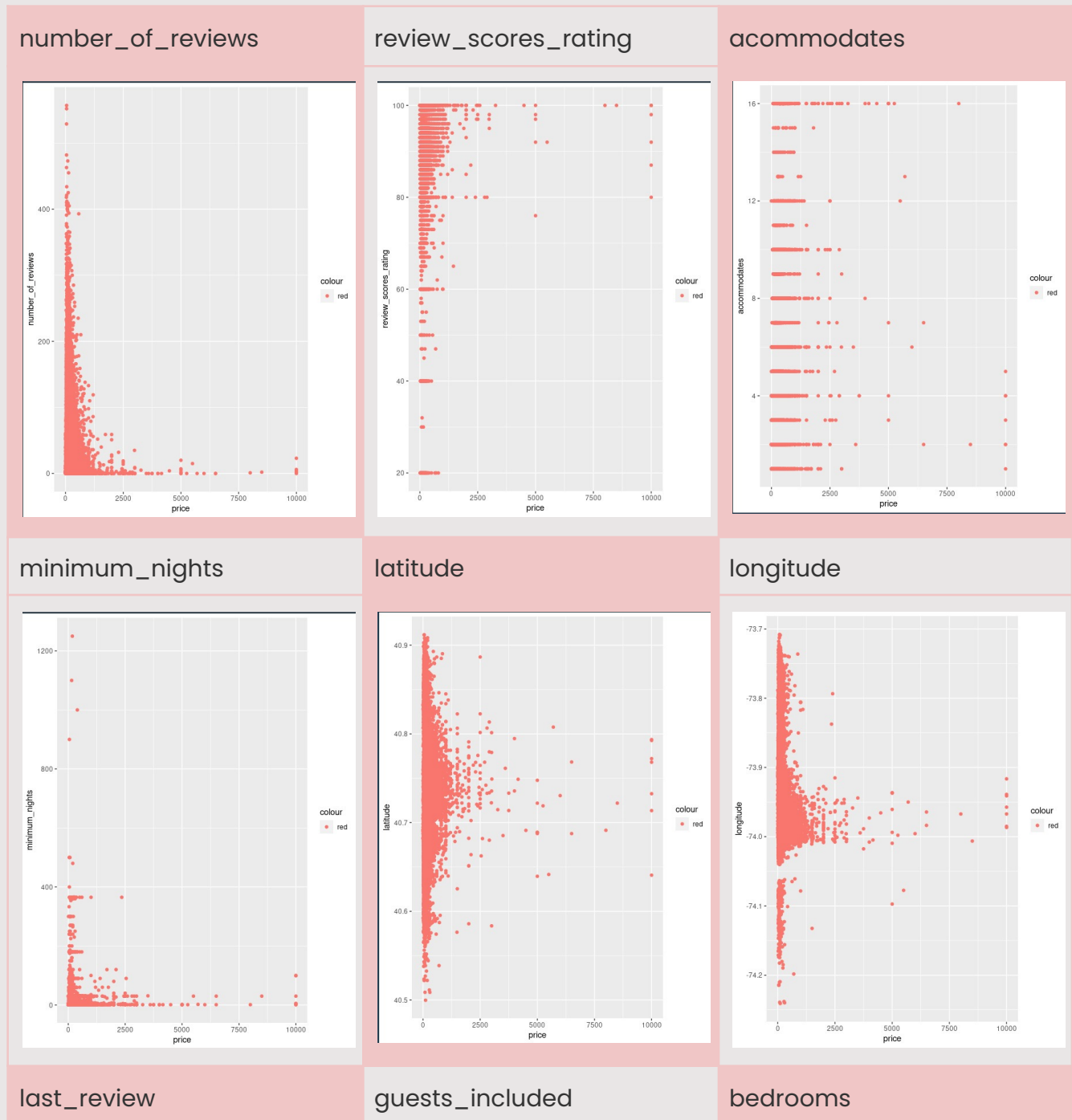


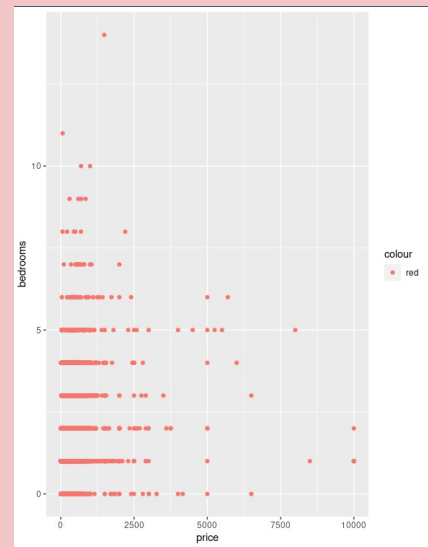
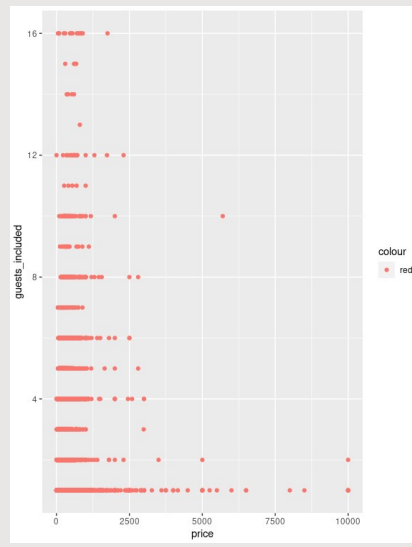
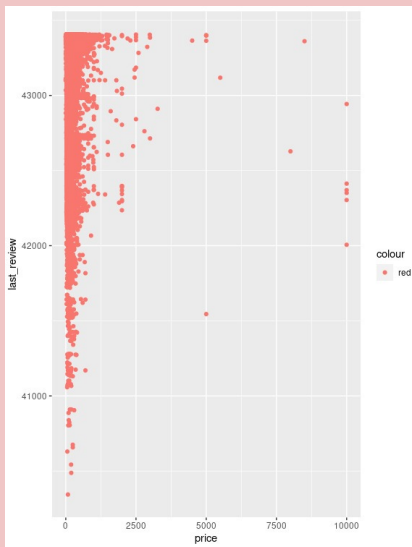
bathrooms



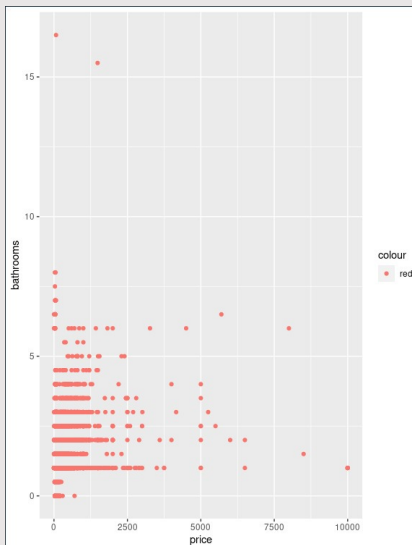
Ejercicio - 2c -

Gráfica de dispersión para cada par de atributos. Deberás realizar una interpretación de los datos de acuerdo con el estudio previo. En esta etapa podrías determinar aspectos como:

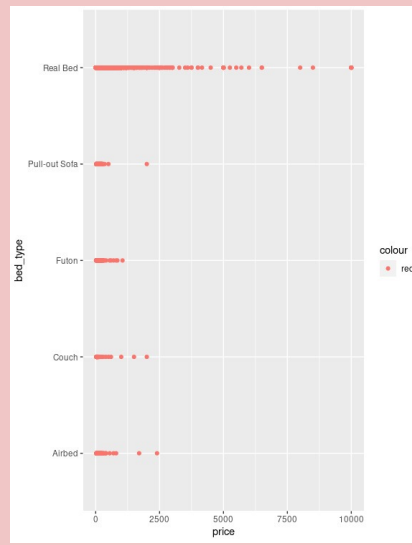




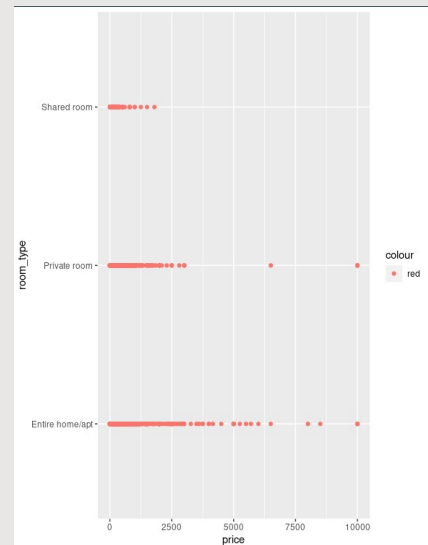
badrooms



bed_type



room_type



property_type

square_feet



- ¿Cuáles atributos parecen estar más ligados a alguna de las variables objetivo del dataset elegido?

Las variables más ligadas a la variable objetivo precio son:

- ☐ Number_Of_Reviews
- ☐ Review_Scores_Rating
- ☐ Guests_Included
- ☐ Bedrooms
- ☐ Bathrooms
- ☐ Bed_type
- ☐ Room_type
- ☐ Property_type
- ☐ Square_Feet

- ¿Cuáles atributos parecen estar menos ligados a alguna de las variables objetivo del dataset elegido?

- ☐ Accomodates
- ☐ Latitude
- ☐ Longitude

- ¿Existen valores correlacionados?

Si, a continuación mostraremos las tablas relacionadas:

- ☐ Latitude - Longitude
- ☐ Property_type - Accomodates
- ☐ Guests_Include - Bedrooms

- Resume en una tabla los hallazgos encontrados, relativos a la predicción de valores de cada atributo.

Number_Of_Reviews	El número de reseñas influye directamente en la predicción del precio, pues mientras más reseñas haya, más clientes habrá y por lo tanto podemos tener un estimado de la ganancia obtenida.
Review_Scores_Rating	La calificación de los puntajes de revisión nos ayudan mucho a la predicción de la ganancia o a la estimación del precio pues sin dicha calificación no habría mayor o menor demanda.
Guests_Included	Se podría decir que es el número de inquilinos o personas que están habitando la propiedad y, en base a eso, se puede establecer cierta renta, lo cual implicaría ganancia y, por tanto, nos ayuda a la predicción del precio.
Bedrooms	Este atributo se escogió porque notamos que el precio aumentaba o disminuía dependiendo del número de dormitorios en una propiedad.

Bathrooms	Los baños influyen directamente en el precio de la renta, alquiler o venta de alguna propiedad pues mientras más baños existan en una propiedad, mayor será el costo de esta y, por tanto, nos ayuda hacer una predicción del precio.
Bed_type	El tipo de cama influye en el precio pues dependiendo del número y tipos de cama (matrimoniales, individuales, etc.), es posible que aumente el precio de una propiedad.
Room_type	El tipo de habitación se escogió porque puede que se tengan habitaciones más chicas o más grandes dependiendo del espacio de la propiedad, esto hace que influya el coste de ella.
Property_type	Para el tipo de propiedad se puede decir que nosotros lo vimos como la zona en donde se encuentra, el espacio de metros con el que cuenta y demás, es por ello que, mientras más grande y mejor situada esté, el precio aumentará.
Square_Feet	El espacio influye directamente en el precio pues mientras más espacio se tenga en la propiedad, más cara saldrá.

- Resume sus hallazgos en una tabla, relativos a la predicción de valores de cada atributo. Indica si existen atributos correlacionados. En caso afirmativo, indica si es posible hacer el proceso de reducción de dimensiones.

Atributos correlacionados	¿Es posible hacer el proceso de reducción de dimensiones?
Latitude - Longitude	En este caso no se puede hacer el proceso de reducción de dimensiones porque los atributos latitud y longitud son muy distintos y se requiere de ambos para encontrar la ubicación exacta de la propiedad.
Property_type - Accommodates	Aquí tenemos una relación estrecha de los atributos porque el número de personas que haya (se acomoden) en una propiedad si depende mucho del tipo de propiedad, más que nada del tamaño, pero ninguno se puede omitir y es por eso que no se puede hacer la reducción de dimensiones.
Guests_Include - Bedrooms	Para estos atributos tampoco se puede hacer la reducción de dimensiones porque, aunque van relacionados el número de inquilinos con los dormitorios, puede que haya casos en donde haya 3 personas durmiendo en una misma habitación. Si se mira de otra manera podemos decir que, el que haya un dormitorio no implica que duerman 3 personas.

Ejercicio - 2d -

Investiga posibles asociaciones de varios atributos con tu variable de clase. Es decir, estudia las gráficas de dispersión elaboradas en el punto anterior y trata de identificar posibles áreas “densas” (si las hay).

Latitud/Longitud: Podemos ver que el área más densamente poblada se encuentra dentro del área que está a menor precio o que, en otras palabras, es más económica.

Property_type: Se puede observar que el mayor número de propiedades se encuentra en la zona más económica; es decir, el tipo de propiedades casi no varía en las zonas de menores recursos.

Number_Of_Reviews: Podemos notar que se cuentan con pocas reseñas y estas pocas se encuentran dentro de las zonas más económicas.

Bathrooms: Se puede notar que las propiedades que cuentan con un número de baños entre 0 y 5 son las propiedades que tienen un coste menor a los 2,500.

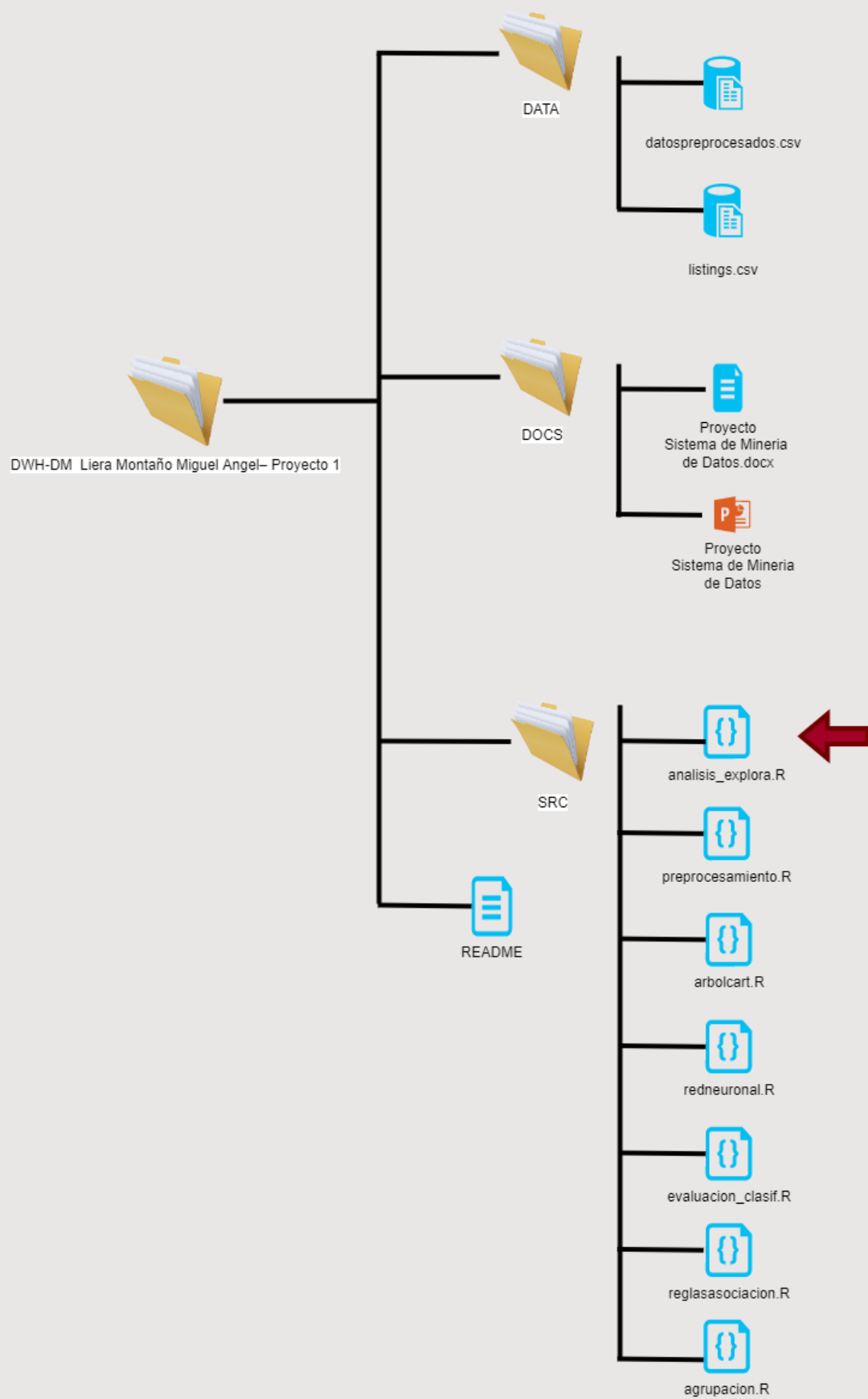
Room_type: Podemos ver que las propiedades que cuentan con el mayor número de dormitorios de distintos tipos son las casas y los apartamentos, aunque el precio varíe.

Guests_Inlcude: Se puede decir que las propiedades con menor número de inquilinos, menor a 4 para ser exactos, son las más habitadas, además se encuentran a un precio bastante accesible.

Review_Scores_Rating: Podemos ver que tenemos un gran número de calificaciones a las propiedades que se encuentran menor a un coste de 2,500

Ejercicio - 2e -

Guardar tu script como analisis_explora.R.



3. Preprocesamiento de datos

En este paso se preparan los datos de acuerdo con las tareas de minería que se van a realizar. Algunos aspectos para considerar son:

Selección de atributos.

Selecciona los atributos que consideres apropiados para una tarea predictiva. Justifica tu respuesta.

ATRIBUTO	
price	Es la variable objetivo
number_of_reviews	Considerado porque si tiene más o menos reviews el precio se ve afectado
review_score_rating	Si tiene buena calificación un lugar tendrá más demanda y el precio será afectado
guest_included	Si hay personas puede que el lugar se vuelva más o menos agradable para los usuarios por lo que se refleja en el precio
bedrooms	Dependiendo de las habitaciones será más caro o barato
bathrooms	Dependiendo del tamaño de la propiedad, debe contar con cierto número de baños si cumple esto puede afectar el precio positivamente
bed_type	Si el lugar cuenta con una buena cama, elevará el precio
room_type	El lugar donde puede estar el usuario afecta el precio

property_type	Si la propiedad es grande o chica tendrá repercusiones en el precio
square_feet	Las dimensiones del lugar afectarán negativa o positivamente al precio

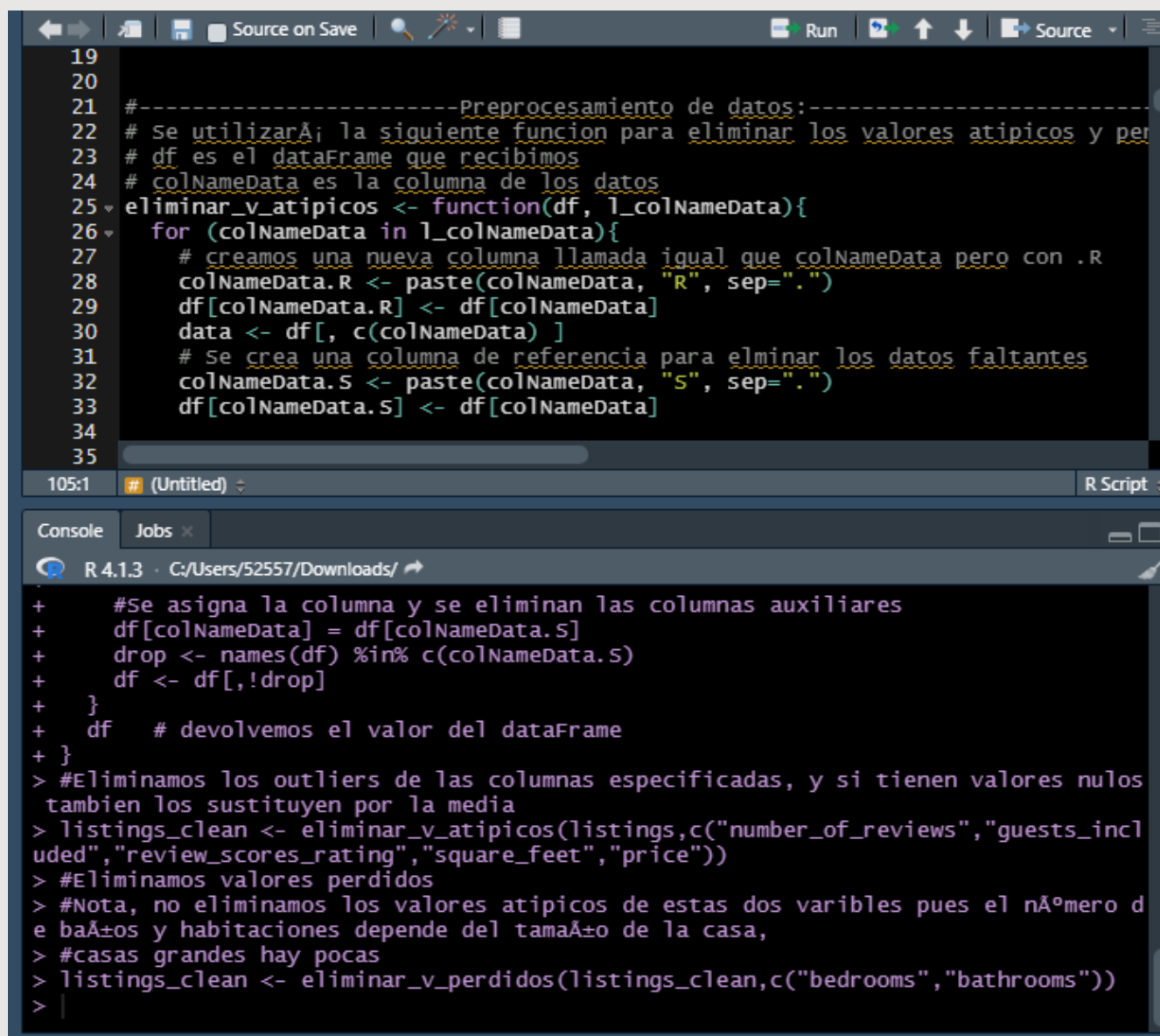
Manejo de valores perdidos.

Considera los siguientes métodos para tratar con valores perdidos:

- Reemplaza los valores perdidos por la media o la moda del atributo, de acuerdo con el tipo de dato del atributo.

ATRIBUTO	Reemplazo de valores perdidos
price	No fue necesario
number_of_reviews	No fue necesario
review_score_rating	Por la media
guest_included	No fue necesario
bedrooms	Por la media
bathrooms	Por la media
bed_type	No fue necesario
room_type	No fue necesario
property_type	No fue necesario
square_feet	Por la media

Anexamos captura de pantalla del código que se implementó para eliminar los valores perdidos y los valores atípicos:



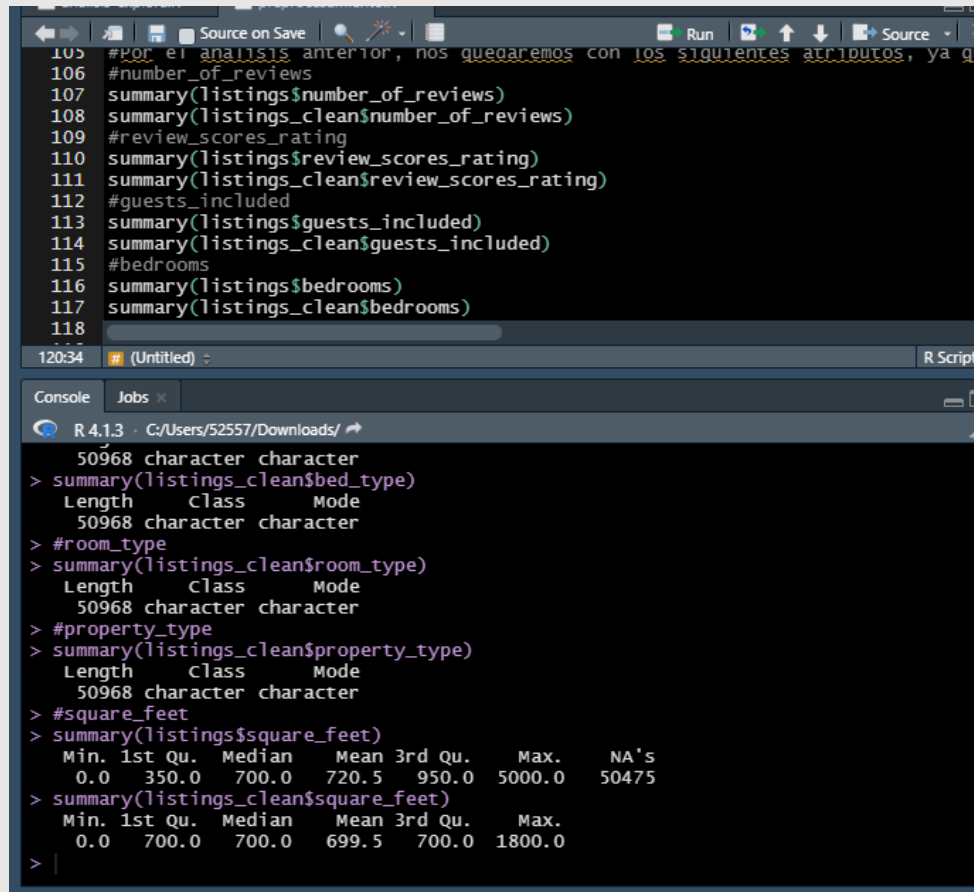
```
19
20
21 #-----Preprocesamiento de datos:-----
22 # Se utilizará la siguiente función para eliminar los valores atípicos y per
23 # df es el dataFrame que recibimos
24 # colNameData es la columna de los datos
25 eliminar_v_atipicos <- function(df, l_colNameData){
26   for (colNameData in l_colNameData){
27     # creamos una nueva columna llamada igual que colNameData pero con .R
28     colNameData.R <- paste(colNameData, "R", sep=".")
29     df[colNameData.R] <- df[colNameData]
30     data <- df[, c(colNameData) ]
31     # Se crea una columna de referencia para eliminar los datos faltantes
32     colNameData.S <- paste(colNameData, "S", sep=".")
33     df[colNameData.S] <- df[colNameData]
34
35
105:1 # (Untitled) R Script

Console Jobs
R 4.1.3 · C:/Users/52557/Downloads/
+ #Se asigna la columna y se eliminan las columnas auxiliares
+ df[colNameData] = df[colNameData.S]
+ drop <- names(df) %in% c(colNameData.S)
+ df <- df[,!drop]
+ }
+ df # devolvemos el valor del dataFrame
+ }
> #Eliminamos los outliers de las columnas especificadas, y si tienen valores nulos
  también los sustituyen por la media
> listings_clean <- eliminar_v_atipicos(listings,c("number_of_reviews","guests_incl
  uded","review_scores_rating","square_feet","price"))
> #Eliminamos valores perdidos
> #Nota, no eliminamos los valores atípicos de estas dos variables pues el número d
  e baños y habitaciones depende del tamaño de la casa,
> #casas grandes hay pocas
> listings_clean <- eliminar_v_perdidos(listings_clean,c("bedrooms","bathrooms"))
> |
```

Eliminación de valores atípicos.

ATRIBUTO	ELIMINACIÓN DE VALORES ATÍPICOS
price	Reemplazados por la media
number_of_reviews	Reemplazados por la media
review_score_rating	Reemplazados por la media
guest_included	No fue necesario eliminar sus valores atípicos
bedrooms	No fue necesario eliminar sus valores atípicos
bathrooms	No fue necesario eliminar sus valores atípicos
bed_type	Reemplazados por la media
room_type	Reemplazados por la media
property_type	Reemplazados por la media
square_feet	Reemplazados por la media

Una vez eliminados los valores atípicos y los valores perdidos, hicimos la comparación entre los datos que se tenían antes con los datos limpios; es decir, sacamos la media, la mediana, mínimos y máximos para ver la diferencia entre los datos.



```
105 #Por el analisis anterior, nos quedaremos con los siguientes atributos, ya q
106 #number_of_reviews
107 summary(listings$number_of_reviews)
108 summary(listings_clean$number_of_reviews)
109 #review_scores_rating
110 summary(listings$review_scores_rating)
111 summary(listings_clean$review_scores_rating)
112 #guests_included
113 summary(listings$guests_included)
114 summary(listings_clean$guests_included)
115 #bedrooms
116 summary(listings$bedrooms)
117 summary(listings_clean$bedrooms)
118
```

120:34 (Untitled) R Script

Console Jobs

R 4.1.3 · C:/Users/52557/Downloads/

```
> 50968 character character
> summary(listings_clean$bed_type)
  Length  Class  Mode
50968 character character
> #room_type
> summary(listings_clean$room_type)
  Length  Class  Mode
50968 character character
> #property_type
> summary(listings_clean$property_type)
  Length  Class  Mode
50968 character character
> #square_feet
> summary(listings$square_feet)
  Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
    0.0   350.0   700.0  720.5   950.0 5000.0 50475
> summary(listings_clean$square_feet)
  Min. 1st Qu. Median Mean 3rd Qu. Max.
    0.0   700.0   700.0  699.5   700.0 1800.0
>
```


Discretización de atributos numéricos.

Convertimos las variables numéricas en categóricas:

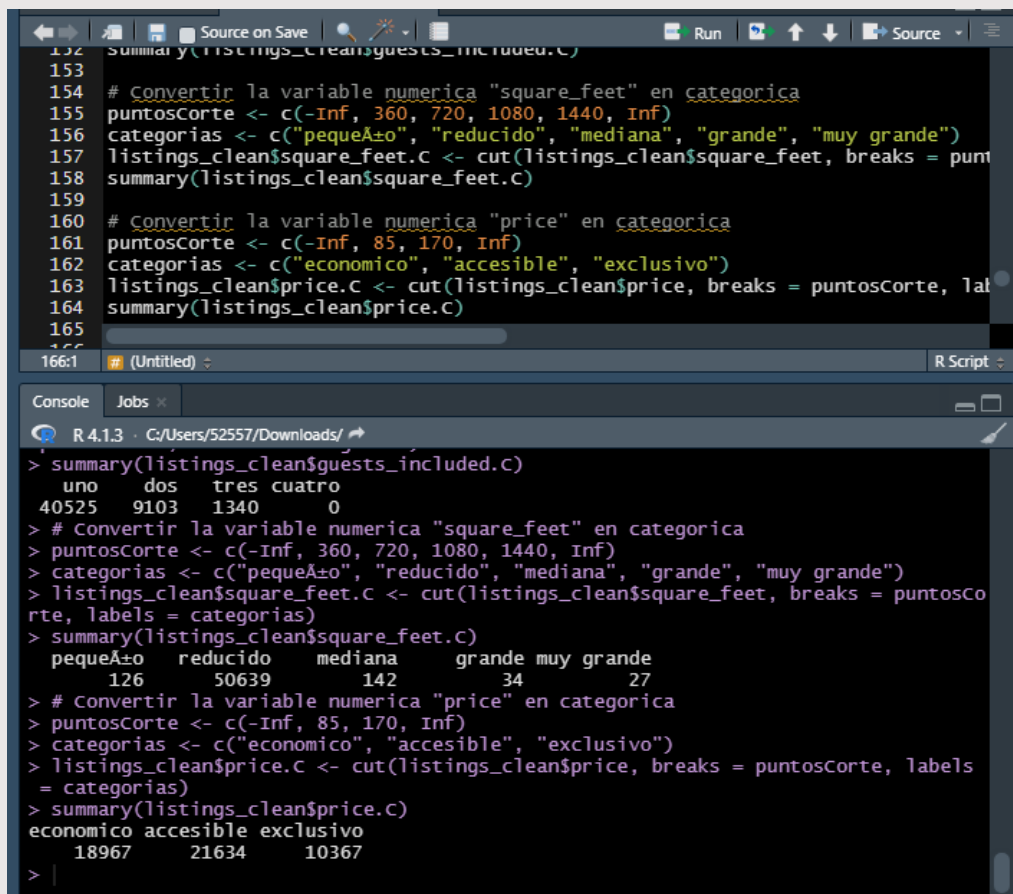
```
# Convertir la variable numerica "number_of_reviews" en categorica
# para ello definimos los puntos de corte
puntosCorte <- c(-Inf, 11, 22, 33, 44, Inf)
categorias_5 <- c("Muy pocas", "Pocas", "Regular", "Varias", "Muchas")
# y cortamos la variable número de pasos segun esta categorizacion
listings_clean$number_of_reviews.C <- cut(listings_clean$number_of_reviews, breaks = puntosCorte, labels = categorias_5)
summary(listings_clean$number_of_reviews.C)

# Convertir la variable numerica "review_scores_rating" en categorica
puntosCorte <- c(70, 85, 90, 95, 100)
categorias <- c("Mala", "Regular", "Buena", "Excelente")
listings_clean$review_scores_rating.C <- cut(listings_clean$review_scores_rating, breaks = puntosCorte, labels = categorias)
summary(listings_clean$review_scores_rating.C)

# Convertir la variable numerica "guests_included" en categorica
puntosCorte <- c(0, 1, 2, 3, Inf)
categorias <- c("uno", "dos", "tres", "cuatro")
listings_clean$guests_included.C <- cut(listings_clean$guests_included, breaks = puntosCorte, labels = categorias)
summary(listings_clean$guests_included.C)

# Convertir la variable numerica "square_feet" en categorica
puntosCorte <- c(-Inf, 360, 720, 1080, 1440, Inf)
categorias <- c("pequeño", "reducido", "mediana", "grande", "muy grande")
listings_clean$square_feet.C <- cut(listings_clean$square_feet, breaks = puntosCorte, labels = categorias)
summary(listings_clean$square_feet.C)

# Convertir la variable numerica "price" en categorica
puntosCorte <- c(-Inf, 85, 170, Inf)
categorias <- c("economico", "accesible", "exclusivo")
listings_clean$price.C <- cut(listings_clean$price, breaks = puntosCorte, labels = categorias)
summary(listings_clean$price.C)
```



The screenshot shows the R Studio environment. The top pane displays R code for discretizing numerical variables. The bottom pane shows the console output for the first three variables: `guests_included.C`, `square_feet.C`, and `price.C`.

```
summary(listings_clean$guests_included.C)
  uno  dos  tres cuatro
40525 9103 1340     0

# Convertir la variable numerica "square_feet" en categorica
puntosCorte <- c(-Inf, 360, 720, 1080, 1440, Inf)
categorias <- c("pequeño", "reducido", "mediana", "grande", "muy grande")
listings_clean$square_feet.C <- cut(listings_clean$square_feet, breaks = puntosCorte, labels = categorias)
summary(listings_clean$square_feet.C)
  pequeño reducido mediana grande muy grande
    126    50639    142     34      27

# Convertir la variable numerica "price" en categorica
puntosCorte <- c(-Inf, 85, 170, Inf)
categorias <- c("economico", "accesible", "exclusivo")
listings_clean$price.C <- cut(listings_clean$price, breaks = puntosCorte, labels = categorias)
summary(listings_clean$price.C)
  economico accesible exclusivo
    18967    21634    10367
```

MÉTODO DE CAMBIO DE VARIABLE NUMÉRICA A CATEGORICA

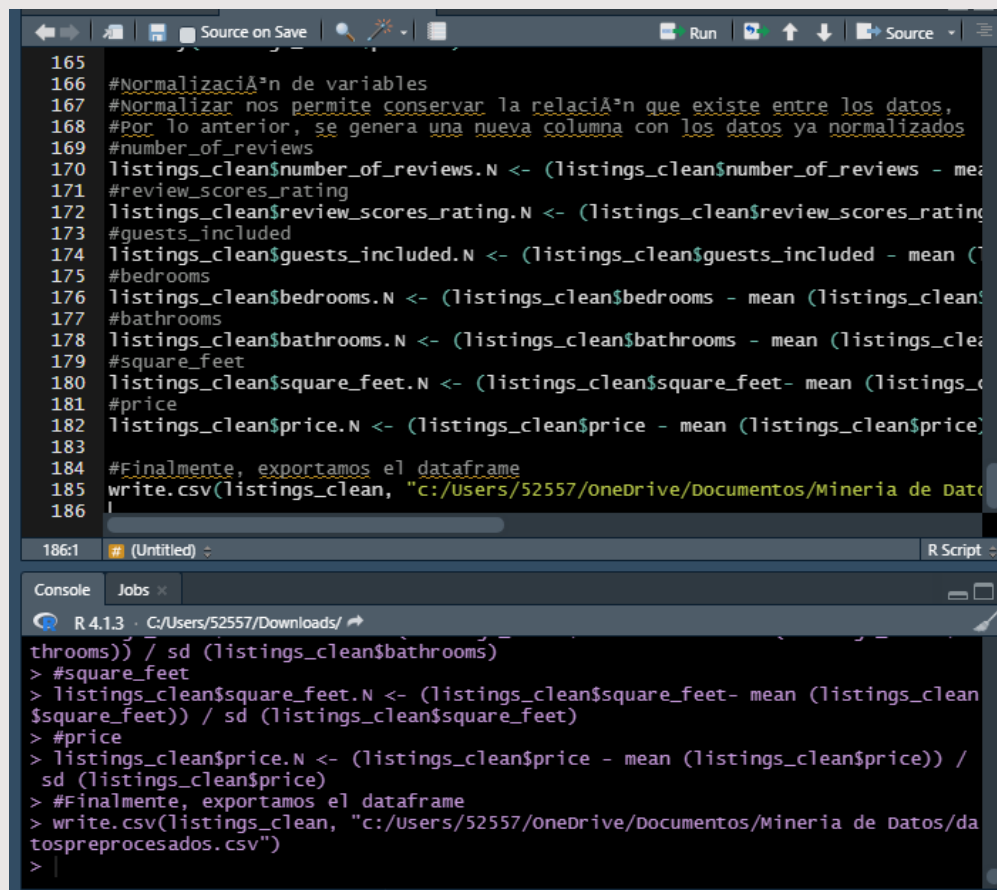
ATRIBUTO	Categorías
price	"económico", "accesible" "exclusivo"
number_of_reviews	"muchas" "varias" "regular" "pocas" "muy pocas"
review_score_rating	"excelente" "buena" "regular" "mala"
guest_included	"1" "2" "3"
bedrooms	ya categorizada
bathrooms	ya categorizada
bed_type	ya categorizada
room_type	ya categorizada
property_type	ya categorizada
square_feet	"muy grande" "grande" "mediana" "reducido" "pequeño"

Normalización.

La normalización es generalmente necesaria cuando se trata de atributos en una escala diferente; de lo contrario, puede conducir a una dilución en la efectividad de un atributo importante igualmente importante (en una escala inferior) debido a que otro atributo tiene valores en una escala mayor. Escalamos los datos de un atributo de modo que caiga en un rango más pequeño, como -1.0 a 1.0 o 0.0 a 1.0 esto será útil, por ejemplo, algoritmos de clasificación.

Anexamos captura de pantalla del código que se llevó a cabo para el proceso de normalización:

```
#Normalización de variables
#Normalizar nos permite conservar la relación que existe entre los datos,
#Por lo anterior, se genera una nueva columna con los datos ya normalizados
#number_of_reviews
listings_clean$number_of_reviews.N <- (listings_clean$number_of_reviews - mean(listings_clean$number_of_reviews)) / sd(listings_clean$number_of_reviews)
#review_scores_rating
listings_clean$review_scores_rating.N <- (listings_clean$review_scores_rating - mean(listings_clean$review_scores_rating)) / sd(listings_clean$review_scores_rating)
#guests_included
listings_clean$guests_included.N <- (listings_clean$guests_included - mean(listings_clean$guests_included)) / sd(listings_clean$guests_included)
#bedrooms
listings_clean$bedrooms.N <- (listings_clean$bedrooms - mean(listings_clean$bedrooms)) / sd(listings_clean$bedrooms)
#bathrooms
listings_clean$bathrooms.N <- (listings_clean$bathrooms - mean(listings_clean$bathrooms)) / sd(listings_clean$bathrooms)
#square_feet
listings_clean$square_feet.N <- (listings_clean$square_feet - mean(listings_clean$square_feet)) / sd(listings_clean$square_feet)
#price
listings_clean$price.N <- (listings_clean$price - mean(listings_clean$price)) / sd(listings_clean$price)
```



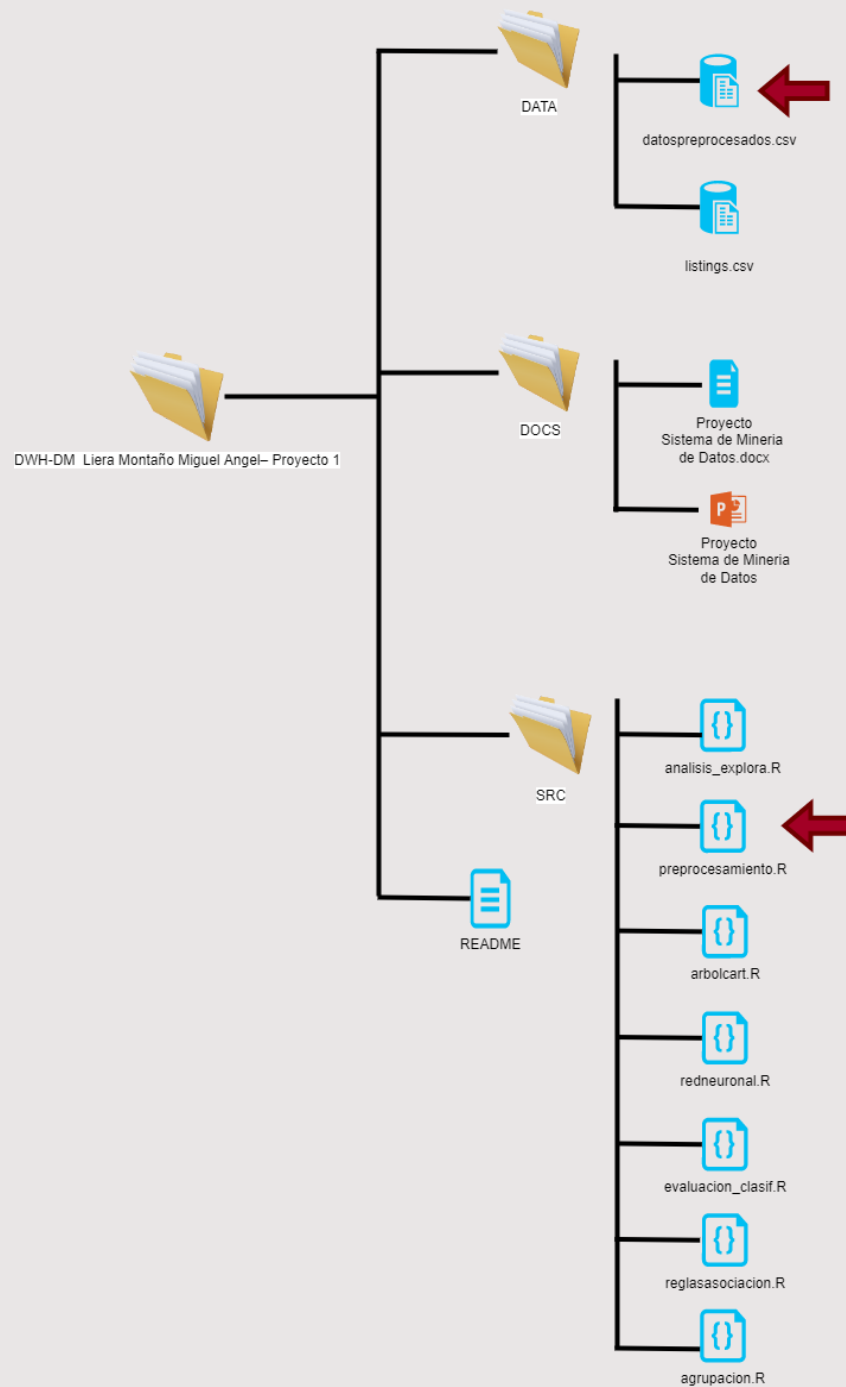
The screenshot shows the R Studio environment. The script editor displays R code for normalizing data. The console shows the execution of the code, including the calculation of normalized values for 'square_feet' and 'price', and the final step of writing the data to a CSV file.

```
165
166 #Normalización de variables
167 #Normalizar nos permite conservar la relación que existe entre los datos,
168 #Por lo anterior, se genera una nueva columna con los datos ya normalizados
169 #number_of_reviews
170 listings_clean$number_of_reviews.N <- (listings_clean$number_of_reviews - mean(listings_clean$number_of_reviews)) / sd(listings_clean$number_of_reviews)
171 #review_scores_rating
172 listings_clean$review_scores_rating.N <- (listings_clean$review_scores_rating - mean(listings_clean$review_scores_rating)) / sd(listings_clean$review_scores_rating)
173 #guests_included
174 listings_clean$guests_included.N <- (listings_clean$guests_included - mean(listings_clean$guests_included)) / sd(listings_clean$guests_included)
175 #bedrooms
176 listings_clean$bedrooms.N <- (listings_clean$bedrooms - mean(listings_clean$bedrooms)) / sd(listings_clean$bedrooms)
177 #bathrooms
178 listings_clean$bathrooms.N <- (listings_clean$bathrooms - mean(listings_clean$bathrooms)) / sd(listings_clean$bathrooms)
179 #square_feet
180 listings_clean$square_feet.N <- (listings_clean$square_feet - mean(listings_clean$square_feet)) / sd(listings_clean$square_feet)
181 #price
182 listings_clean$price.N <- (listings_clean$price - mean(listings_clean$price)) / sd(listings_clean$price)
183
184 #Finalmente, exportamos el dataframe
185 write.csv(listings_clean, "c:/Users/52557/OneDrive/Documentos/Mineria de Datos/datospreprocesados.csv")
186
```

Console output:

```
R 4.1.3 · C:/Users/52557/Downloads/
> listings_clean$bathrooms.N <- (listings_clean$bathrooms - mean(listings_clean$bathrooms)) / sd(listings_clean$bathrooms)
> #square_feet
> listings_clean$square_feet.N <- (listings_clean$square_feet - mean(listings_clean$square_feet)) / sd(listings_clean$square_feet)
> #price
> listings_clean$price.N <- (listings_clean$price - mean(listings_clean$price)) / sd(listings_clean$price)
> #Finalmente, exportamos el dataframe
> write.csv(listings_clean, "c:/Users/52557/OneDrive/Documentos/Mineria de Datos/datospreprocesados.csv")
>
```

Una vez finalizado este proceso, deberás guardar el dataset resultante en un archivo con el nombre de datos preprocesados.csv. También deberás guardar el script en que te apoyaste para realizar las tareas de preprocesamiento. Guarda tu script como preprocesamiento.R.



4. Minado de datos

Repetir los pasos descritos abajo para el dataset creado en el punto anterior.

Utiliza un árbol CART.

Utiliza diferentes valores para parámetros tales como podado, cp y/o cantidad mínima de registros en las hojas. Describe los patrones obtenidos y compáralos con las conclusiones previas. Evalúa tus resultados (ve más allá de la exactitud). Describe los patrones obtenidos y compáralos con las conclusiones previas. Guardar tu script como arbolcart.R.

Se particionó el data frame original en entrenamiento y prueba, y se eliminaron algunos valores del atributo property_type

```
#Creamos los conjuntos de entrenamiento y prueba, particionando el dataset original
listings_entrenamiento <- sample_frac(listings_clean, .7)
listings_prueba <- setdiff(listings_clean, listings_entrenamiento)
listings_prueba <- listings_prueba[listings_prueba$property_type != "Nature lodge" &
                                   listings_prueba$property_type != "Timeshare" & listings_prueba$property_type != "Train", ]
```

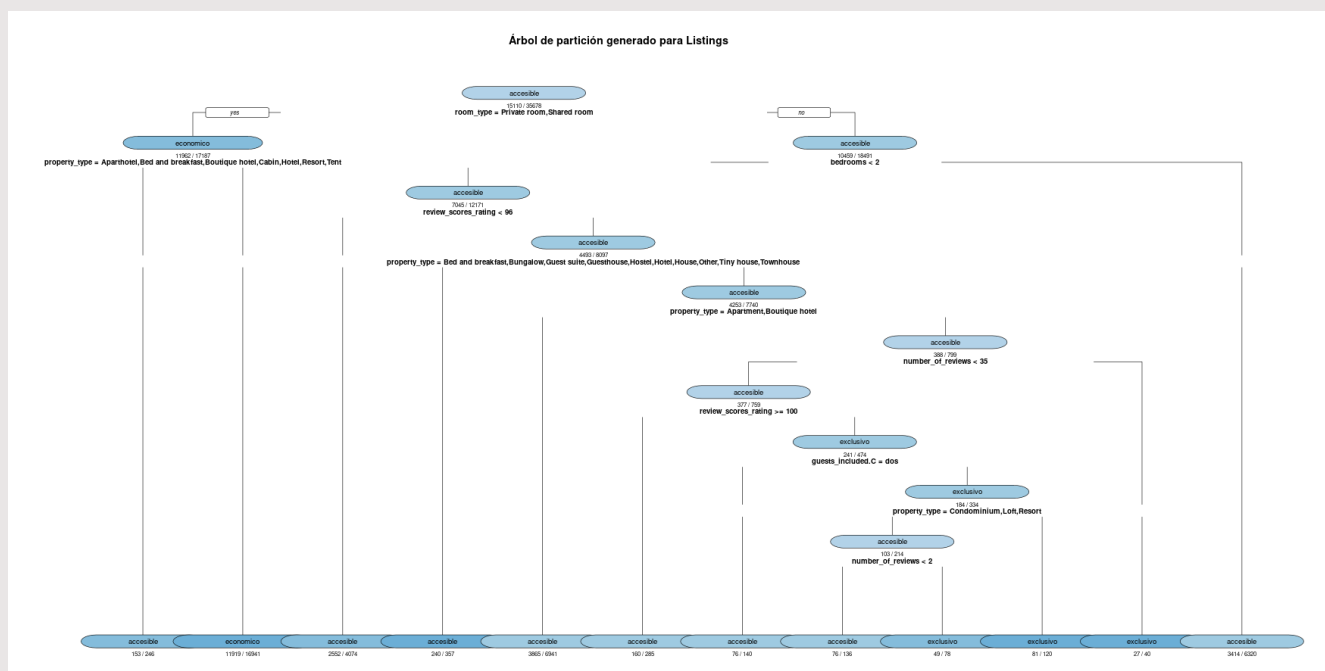
Se creó el modelo de la siguiente forma:

```
#0.000188
#Se muestra el árbol generado
rpart.plot(arbol_1, box.palette="Blues", main="Árbol de partición generado para Listings",extra=2,under=TRUE,varlen=0,facilen=0,cex=.6)

#Se muestran las especificaciones del árbol
printcp(arbol_1)

#Generamos la predicción
prediccion_1 <- predict(arbol_1, listings_prueba, type="class")
```

Se optó por cambiar el parámetro control = rpart.control(cp=0.000325, minbucket = 0). Para que la profundidad del árbol sea la correcta, y la cantidad de tuplas mínimas existentes en un nodo sea 0 (esto generaría un árbol enorme, lo cual es compensado por el parámetro cp). La arquitectura del árbol obtenido es la siguiente:



La salida del árbol modelado fue el siguiente:

Classification tree:

```
rpart(formula = price.C ~ (number_of_reviews + review_scores_rating +
  guests_included + bedrooms + bed_type + room_type + property_type +
  square_feet.C + guests_included.C + number_of_reviews.C +
  review_scores_rating.C), data = listings_entrenamiento, method = "class",
  control = rpart.control(cp = 0.00032, minbucket = 0))
```

Variables actually used in tree construction:

```
[1] bedrooms          guests_included.C  number_of_reviews  property_type      review_scores_rating room_type
```

Root node error: 20568/35678 = 0.57649

n= 35678

	CP	nsplit	rel error	xerror	xstd
1	0.35545508	0	1.00000	1.00000	0.0045377
2	0.00534811	1	0.64454	0.64454	0.0044377
3	0.00032413	2	0.63920	0.63983	0.0044310
4	0.00032000	11	0.63526	0.63900	0.0044298

La matriz de confusión obtenida para evaluar los resultados predictivos del árbol es el siguiente:

```

Confusion Matrix and Statistics

              Reference
Prediction  accesible economico exclusivo
accesible    4551         668       2865
economico    1923        4924        247
exclusivo      47          2         60

Overall Statistics

              Accuracy : 0.6237
              95% CI : (0.616, 0.6314)
              No Information Rate : 0.4266
              P-Value [Acc > NIR] : < 2.2e-16

              Kappa : 0.3761

              Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

              Class: accesible Class: economico Class: exclusivo
Sensitivity              0.6979              0.8802              0.018916
Specificity              0.5970              0.7761              0.995955
Pos Pred Value           0.5630              0.6941              0.550459
Neg Pred Value           0.7265              0.9182              0.794966
Prevalence               0.4266              0.3659              0.207497
Detection Rate           0.2977              0.3221              0.003925
Detection Prevalence     0.5288              0.4641              0.007130
Balanced Accuracy        0.6474              0.8282              0.507435
>

```

Como vemos, la precisión del árbol es de 0.62, que es baja pero aceptable, ya que es mejor que el azar

Red Neuronal

Utiliza diferentes valores para parámetros tales como momentum, tasa de aprendizaje, número de épocas, cantidad de capas ocultas y/o número de nodos en ellas (siempre que la herramienta lo permita). Evalúa tus resultados. Describe los patrones obtenidos y compáralos con las conclusiones previas. Guardar tu script como redneuronal.R.

Como se sabe, una red neuronal consiste en un modelo de clasificación o regresión compuesto de unidades de procesamiento (neuronas) que en conjunto resuelven una problemática. Ante esto la creación de un modelo de red para predecir nuestra variable objetivo resulta bastante útil, por lo que se implementaron algunos cambios en el dataset

para que el modelo de red pudiera recibir como entradas a las variables predictivas y como objetivo a precio. A continuación se muestra el preprocesamiento realizado:

```
#Con el objetivo de reconvertir las variables discretizadas anteriormente a valores numéricos para que la topología de la red las tome como variables predictivas,
#se crean nuevas discretizaciones con números para identificar cada una

# Convertir la variable numerica "number_of_reviews" en categorica
puntosCorte <- c(-Inf, 11, 22, 33, 44, Inf)
categorias <- c("0", "1", "2", "3", "4")
listings_clean$number_of_reviews.CN <- as.numeric(cut(listings_clean$number_of_reviews, breaks = puntosCorte, labels = categorias))

# Convertir la variable numerica "review_scores_rating" en categorica
puntosCorte <- c(70, 85, 90, 95, 100)
categorias <- c("0", "1", "2", "3")
listings_clean$review_scores_rating.CN <- as.numeric(cut(listings_clean$review_scores_rating, breaks = puntosCorte, labels = categorias))
summary(listings_clean$review_scores_rating.CN)

# Convertir la variable numerica "guests_included" en categorica
puntosCorte <- c(0, 1, 2, 3, Inf)
categorias <- c("0", "1", "2", "3")
listings_clean$guests_included.CN <- as.numeric(cut(listings_clean$guests_included, breaks = puntosCorte, labels = categorias))
summary(listings_clean$guests_included.CN)

# Convertir la variable numerica "square_feet" en categorica
puntosCorte <- c(-Inf, 360, 720, 1080, 1440, Inf)
categorias <- c("0", "1", "2", "3", "4")
listings_clean$square_feet.CN <- as.numeric(cut(listings_clean$square_feet, breaks = puntosCorte, labels = categorias))
summary(listings_clean$square_feet.CN)

# Convertir la variable numerica "price" en categorica
puntosCorte <- c(-Inf, 85, 170, Inf)
categorias <- c("0", "1", "2")
listings_clean$price.CN <- as.numeric(cut(listings_clean$price, breaks = puntosCorte, labels = categorias))
summary(listings_clean$price.CN)

#Convertimos la variable categorica "bed_type" en numerica
levels <- c("Airbed","Couch","Futon","Pull-out Sofa","Real Bed")
listings_clean$bed_type.CN <- match(listings_clean$bed_type, levels)

#Convertimos la variable categorica "room_type" en numerica
levels <- c("Entire home/apt","Private room","Shared room")
listings_clean$room_type.CN <- match(listings_clean$room_type, levels)
```

Como observamos, los atributos anteriormente ya categorizados se convirtieron en numéricos nuevamente, a manera que cada numero representara una categoría de cada atributo para que la red pudiera recibir como entradas a estos valores.

```
#Hacemos un un one hot encoding para las variables objetivo:

listings_tmp <- dummyVars(" ~ price.C", data = listings_RN)
listings_tmp_f <- data.frame(predict(listings_tmp, newdata = listings_clean))

#Asignamos las columnas a nuestro conjunto de datos
listings_RN$price.C.economico <- unlist(listings_tmp_f[1])
listings_RN$price.C.accesible <- unlist(listings_tmp_f[2])
listings_RN$price.C.exclusivo <- unlist(listings_tmp_f[3])

listings_RN_entrenamiento <- sample_frac(listings_RN, .66)
listings_RN_prueba <- setdiff(listings_RN, listings_RN_entrenamiento)

#creamos el modelo
```


Uno de los procesos que más podemos destacar es hacer uso de la famosa técnica One Hot Encoding, que consiste en crear una salida en la red evaluando a un solo atributo, de manera que existirán tantas salidas en la red neuronal como valores que tenga la variable a predecir. En este caso price posee tres valores posibles: “económico”, “accesible” y “exclusivo” por lo que el modelo de la red tendrá como salida 3 neuronas, cada una representando a cada uno de estos valores posibles. Para lograr lo anterior, fue necesario la creación de las tres nuevas columnas, que hacen referencia a si una tupla es o no evaluada a cierta categoría. A continuación se muestra el conjunto de datos resultante:

Se modela así por el tamaño de los datos para que pueda converger a una solución no podíamos poner 70 u 80 neuronas porque el tiempo de entrenamiento se extendería por el peso de la entrada.

Nota: Hubo un problema que se nos presentó a la hora de ejecutar el entrenamiento de la red, ya que al tratarlo de llevar a cabo saltaba el siguiente mensaje:

```
> red <- neuralnet((price.C.economico + price.C.accesible + price.C.exclusivo) ~ (bedrooms +  
+ review_scores_rating.CN + square_feet.CN + bed_type.CN + room_type.CN + property_type.CN),  
+ listings_RN_entrenamiento, hidden=c(8,12,14,8), linear.output = FALSE, stepmax = 2000)  
Warning message:  
Algorithm did not converge in 1 of 1 repetition(s) within the stepmax.
```

Este mensaje implicaba que el algoritmo de entrenamiento no tuvo el tiempo suficiente para que la red converga a una respuesta correcta. La solución más lógica a este problema fue aumentar el número atribuido al parámetro stepmax, de manera que la red tuviera el tiempo y recursos necesarios para, si no dar un resultado inmediatamente correcto, arrojar la información necesaria para modificar el modelo y lograr nuestro objetivo.

Estuvimos corriendo el script alrededor de 2 horas, ya que cuando tardaba menos tiempo, es decir, cuando el parámetro stepmax era inicializado con un valor menor, se lanzaba el mismo error antes mencionado. Sin embargo, la red no arrojó resultados, por lo que decidimos dejar el código como estaba.

Evaluación de modelos de clasificación.

Derivado de los puntos a y b construiste dos clasificadores. Necesitas evaluar la calidad de los modelos y compararlos. Resume en una tabla las diferentes medidas de

evaluación de cada clasificador. ¿Qué clasificador resulta mejor? ¿Por qué? Guardar tu script como `evaluacion_clasif.R`.

¿Qué puedes concluir?

Llegamos a la conclusión de que la complejidad de un árbol de clasificación es mucho menor que el de la red neuronal debido al tiempo que se tarda en cargar. Además el algoritmo de entrenamiento del árbol es más autónomo porque solo se modifica el CP y el Mini Pocket para llegar a un resultado medianamente certero, mientras que con la Red Neuronal se tendría que probar con diferente cantidad de neuronas y capas ocultas.

¿Qué clasificador resulta mejor?

Definitivamente podemos decir que Árbol Cart, es mucho más eficaz debido a la reducción de tiempo que ofrece, además de la facilidad que tiene para manejarlo.

Utiliza reglas de asociación.

Usa reglas de asociación para construir reglas de alta confianza para predecir la variable objetivo que hayas definido Usa el método A priori y describe los patrones obtenidos, comparándolos con las conclusiones previas. Haz de ser posible una comparación de los resultados obtenidos con el otro método revisado en clase. Guardar tu script como `reglasasociacion.R`.

Tenemos la preparación de datos para implementar el algoritmo a priori. Se toma una fracción del dataset para crear el conjunto de transacciones, ya que al momento de implementar el algoritmo la generación de combinaciones para el lado izquierdo de las reglas será enorme (En contadas ocasiones RStudio nos mostró una ventana emergente con un aviso que informaba sobre un error en la sesión).

```
#Reglas de asociacion
library(tidyverse)
library(arules)

#leemos el .csv
listings<- read.csv("~/Documentos/Almacenes y Minería de Datos/Proyecto_Final/Sistema-de-Minería-de-Datos/datospreprocesados.csv")

#Nota: Se usará una fracción reducida del conjunto de datos para generar las transacciones y reglas para evitar que el conjunto de estas últimas
#genere una explosión combinatoria
listings_m <- sample_frac(listings, .1)
listings <- listings_m
dim(listings)
#listings_RN_prueba <- setdiff(listings_RN, listings_RN_entrenamiento)

tid <- as.character(listings[["id"]])
listings <- listings[, -1]

#Hacemos que todas las columnas sean de tipo factor
for(i in 1:ncol(listings))listings[[i]]<-as.factor(listings[[i]])
trans <- as(listings, "transactions")

#Establecemos el ID de la transacción
transactionInfo(trans)[["transactionID"]] <- tid
inspect(trans[1:5])
```

La confianza y el soporte fueron establecidas de manera similar a los ejemplos vistos en clase.

Notemos que la longitud mínima para las reglas de asociación es de 5, de esta manera garantizamos que la regla abarque la mayor cantidad de atributos predictivos, y no se limite a generar transacciones con un solo elemento en el lado izquierdo que solamente entorpecieron el análisis:

```
# Reglas de asociación

#Calcula las reglas de asociación a partir del algoritmo apriori, con una confianza del 70%
# Se especifica en los parametros de la parte derecha de cada regla que los únicos valores permitidos
#son los de la variable discretizada de precio "price.C"
soporte <- 30 / dim(trans)[1]
reglas <- apriori(data = trans,
  parameter = list(support = soporte,
    minlen = 5, #De esta manera garantizamos que la mayoría de atributos se vean implicados en las reglas
    confidence = 0.70,
    # Se especifica que se creen reglas
    target = "rules"),
  appearance = list(rhs = c("price.C=economico", "price.C=accesible", "price.C=exclusivo"),
    lhs=c("number_of_reviews.C=Muchas", "number_of_reviews.C=Muy pocas", "number_of_reviews.C=Pocas", "number_of_r
      "bathrooms=0", "bathrooms=1", "bathrooms=3", "bathrooms=4", "bathrooms=6", "bedrooms=1", "bedrooms=2", "bedro
      "bed_type=Airbed", "bed_type=Couch", "bed_type=Futon", "bed_type=Real Bed", "bed_type=Pull-out Sofa", "prop
      "property_type=Serviced apartment", "property_type=Villa",
      "room_type=Entire home/apt", "room_type=Private room", "room_type=Shared room", "review_scores_rating.C=8
      "review_scores_rating.C=Mala", "review_scores_rating.C=Regular", "guests_included=1", "guests_included=2",
      "square_feet.C=Muy grande", "square_feet.C=pequeño", "square_feet.C=reducido"))))

#Nos da información general de las reglas, incluyendo los valores de promedio, minimo, maximo, mediana, etc.
# del soporte, confianza, cobertura, y otras medidas de las reglas
summary(reglas)

#Nos devuelve un desglose completo de las reglas ordenadas en orden decreciente de confianza,
#esto incluye el soporte, la confianza, cobertura, lift y frecuenciencia
#De esta manera obtenemos las reglas con mayor confianza que nos son útiles para obtener nuestra variable objetivo
inspect(sort(x = reglas, decreasing = TRUE, by = "confidence")[1:20])
```

Con la ejecución del código anterior, obtenemos la siguiente salida:

```
Apriori

Parameter specification:
confidence minval smax arem aval originalSupport maxtime      support minlen maxlen target ext
      0.7   0.1   1 none FALSE      TRUE       5 0.005885815     5    10 rules TRUE

Algorithmic control:
filter tree heap memopt load sort verbose
  0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 30

set item appearances ...[46 item(s)] done [0.00s].
set transactions ...[46 item(s), 5097 transaction(s)] done [0.03s].
sorting and recoding items ... [29 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10 done [0.01s].
writing ... [698 rule(s)] done [0.00s].
creating 54 object ... done [0.00s].
>
> #Nos da información general de las reglas, incluyendo los valores de promedio, mínimo, máximo, mediana, etc.
> # del soporte, confianza, cobertura, y otras medidas de las reglas
> summary(reglas)
set of 698 rules

rule length distribution (lhs + rhs):sizes
  5   6   7   8   9  10
160 226 190  93  26   3

      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      5.000   6.000   6.000   6.438   7.000  10.000

summary of quality measures:
      support      confidence      coverage      lift      count
Min.   :0.005886 Min.   :0.7000 Min.   :0.007259 Min.   :1.870 Min.   : 30.00
1st Qu.:0.008633 1st Qu.:0.7204 1st Qu.:0.010791 1st Qu.:1.925 1st Qu.: 44.00
Median :0.011183 Median :0.7533 Median :0.014322 Median :2.012 Median : 57.00
Mean   :0.014742 Mean   :0.7603 Mean   :0.019807 Mean   :2.031 Mean   : 75.14
3rd Qu.:0.017412 3rd Qu.:0.7959 3rd Qu.:0.023445 3rd Qu.:2.126 3rd Qu.: 88.75
Max.   :0.260742 Max.   :0.8519 Max.   :0.371591 Max.   :2.276 Max.   :1329.00

mining info:
      data ntransactions      support confidence
trans      5097 0.005885815      0.7
```

```
> #Nos devuelve un desglose completo de las reglas ordenadas en orden decreciente de confianza,
> #esto incluye el soporte, la confianza, cobertura, lift y frecuencia
> #De esta manera obtenemos las reglas con mayor confianza que nos son útiles para obtener nuestra variable objetivo
> inspect(sort(x = reglas, decreasing = TRUE, by = "confidence")[1:20])
      lhs                                rhs      support confidence      coverage      lift count
[1] {room_type=Shared room,
      bedrooms=1,
      guests_included=1,
      square_feet.C=reducido}      => {price.C=economico} 0.018049833  0.8518519 0.021188935 2.275623   92
[2] {room_type=Shared room,
      bedrooms=1,
      guests_included=1,
      number_of_reviews.C=Muy pocas} => {price.C=economico} 0.014322150  0.8488372 0.016872670 2.267570   73
```

Como podemos observar, la confianza máxima que se alcanza es de a lo sumo 87% (obtenido en otras ejecuciones). Para realizar la comparación, se emplea la ayuda del método ruleInduction

```

#Como vemos, la mayor confianza que se tiene es del 87%, vamos a comparar con el metodo de ruleInduction

closed <- apriori(trans,
                  parameter = list(target = "closed", minlen = 5, support = soporte, confidence = 0.7))

reglas_t <- ruleInduction(closed, trans, verbose = TRUE)

#Nos da información general de las reglas
summary(reglas_t)

#Obtenemos las reglas con mayor confianza que nos son útiles para obtener nuestra variable objetivo
inspect(sort(x = reglas_t, decreasing = TRUE, by = "confidence")[1:20])

```

Utiliza agrupación.

Investiga si hay una tendencia de agrupamiento en el dataset. Empieza agrupando los datos con el algoritmo k-medias para segmentar los registros en grupos similares. Perfila los clústeres, es decir, qué se puede aprender del tipo de registros que hay en cada clúster. Descríbelo en español. Encuentra un valor adecuado para k. Justifica tu respuesta. Usa el atributo de clase para evaluar el clúster y asegúrate que la desviación estándar se calcule sobre los atributos numéricos. Obtén conclusiones de las medidas numéricas desplegadas para cada clúster. Guardar tu script como agrupacion.R.

5. Conclusiones

Podemos decir que definitivamente es necesaria una limpieza de datos en una empresa/compañía debido a que hay situaciones en las que nos encontramos con datos que son irrelevantes para el manejo de la misma o que incluso hay datos que tienen que llenarse con otros para poder hacer un análisis.

De igual manera pudimos observar que, al menos en nuestra variable objetivo, hay muchísimos otros datos que influyen de manera positiva o negativa en ella (en el precio); a su vez, nos encontramos con otros datos que nos ayudaron a la predicción o estimación de nuestra misma variable objetivo.

Nos dimos cuenta de que, a pesar de no ser muy fanáticos de las gráficas e histogramas, nos fueron de mucha utilidad para la lectura de los datos.

Podemos agregar que la parte final del trabajo fue sumamente difícil pues, como eran muchos datos, la red neuronal nos tardó muchísimo en cargar y eso hizo que perdiéramos demasiado tiempo en el análisis, es por eso que preferimos el árbol cart. También nos dimos cuenta de que fue necesario investigar un poco más allá del proyecto, para entender el manejo de la empresa dada y poder hacer un manejo más eficiente y realista de los datos.