UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Proyectos II, integración, calidad y análisis exploratorio de datos

Segunda presentación:

Integración de datos

Introducción

Antes de cada HITO de presentación, debemos rellenar estas fichas y presentarlas a través de PoliformaT, en la tarea que indiquen los profesores. Cada equipo de trabajo presenta las mismas fichas. Sólo será necesario que las suba uno de los componentes del equipo.

Nombres y apellidos de los autores:

Nadal	Bardisa Quintero
Antonio	Sánchez Guillén
Isabelle	Archer
María	Martínez Fernández
Sergio	Vidal Picazo

2. Interés y alcance del proyecto.

Contesta a las preguntas que plantea este formulario.

2.1. Explica el objetivo principal de tu proyecto ¿qué presenta este estudio?

El objetivo principal del proyecto es estudiar la vivienda en oferta de Airbnb.

Pretendemos descubrir el efecto que tiene cada variable a nuestro alcance en el precio de las viviendas ofertadas en alquiler por la empresa Airbnb. Somos conscientes de la importante repercusión que puede tener la obtención de esta información y queremos darla a conocer tanto a usuarios corrientes como a cualquier órgano con poder para mejorar el mercado de la vivienda de alquiler en España.

Conocer qué variables hacen aumentar o disminuir el precio de la vivienda resulta interesante para posteriormente realizar predicciones de precio, definiendo así viviendas que se encuentren por debajo de este como "gangas" o "chollos" y retratando las que se encuentren por encima como "timos" o "robos" para los consumidores corrientes.

2.2. Explica para qué y para quién podría ser de utilidad este estudio

Este análisis que llevamos a cabo en el proyecto será de gran utilidad tanto para usuarios de Airbnb a la hora de buscar un apartamento donde alojarse en base a ciertas variables (para encontrar un apartamento de las características deseadas y al mejor precio) como para gente externa que esté interesada en la evolución de los precios de la vivienda en alquiler, por ejemplo, a otros analistas que pretendan encontrar relaciones entre los precios de las viviendas en alquiler respecto de las viviendas en venta.

Resultaría realmente útil a los órganos de gobierno autonómicos para entender los problemas en el sector de la vivienda desde el punto de vista de los ciudadanos, permitiendo así abordarlos de manera más efectiva y eficiente y, por ende, mejorando la

calidad de vida de los ciudadanos y aumentando el nivel de felicidad general y aceptación hacia el grupo gobernante del momento.

2.3. ¿Por qué piensas que es novedoso? ¿has visto estudios similares?

Consideramos que, pese a existir muchos análisis de los precios de las viviendas tanto en Valencia como en el resto de España, nuestro análisis puede resultar de interés al tener en cuenta únicamente las viviendas que se ofrecen en alquiler por tiempo limitado disponibles en Airbnb. Además, buscamos encontrar relaciones que no se hayan podido tener en cuenta en análisis anteriores por haber sido consideradas marginales o poco útiles en las estimaciones. También hay que tener en cuenta que, con los cambios de gobierno y políticas de vivienda, los precios pueden verse afectados de nuevas formas, tomando relevancia nuevas variables o incluso eliminando otras para las estimaciones actuales. Además, vamos a realizar un análisis de sentimiento de las reseñas de cada vivienda, que pueden influir o no en el precio. También puede usarlo para ver si el precio depende más de la ubicación en la vivienda en sí.

2.4. Alcance (objetivos definitivos del proyecto)

Define los objetivos del análisis de datos de tu proyecto. Se deben presentar 5 objetivos de análisis.

- Estudiar diferencias de precios entre viviendas ubicadas en diferentes zonas.
- Analizar la relación entre la fiabilidad del propietario y la disponibilidad de la vivienda.
- Valorar el efecto de las reseñas positivas y negativas en el precio de la vivienda y comparar las reseñas de inquilinos hispanohablantes y no hispanohablantes para identificar diferencias en percepción, expectativas y satisfacción con la vivienda.
- Análisis de la relación de renta media de los hogares de una zona con el precio de alquiler en esa zona.
- Predicción del precio del alquiler de la ciudad (genérico) y por zona (barrio o distrito) de la ciudad para comparar que modelo es mejor.

3. Calidad y Análisis exploratorio.

Describe la calidad de los datos y los resultados del análisis exploratorio efectuado. Explica el trabajo técnico, como, por ejemplo, estadísticas aplicadas, visualizaciones o representaciones que has utilizado (puedes poner alguna captura como ejemplo), etc. Valora es esfuerzo del análisis exploratorio de tu proyecto.

No se trata de describir los objetivos resultado del proyecto, sino lo que has tenido que hacer para entender los datos, ver qué nos ofrecen, cómo de "sucia" es la fuente, etc.

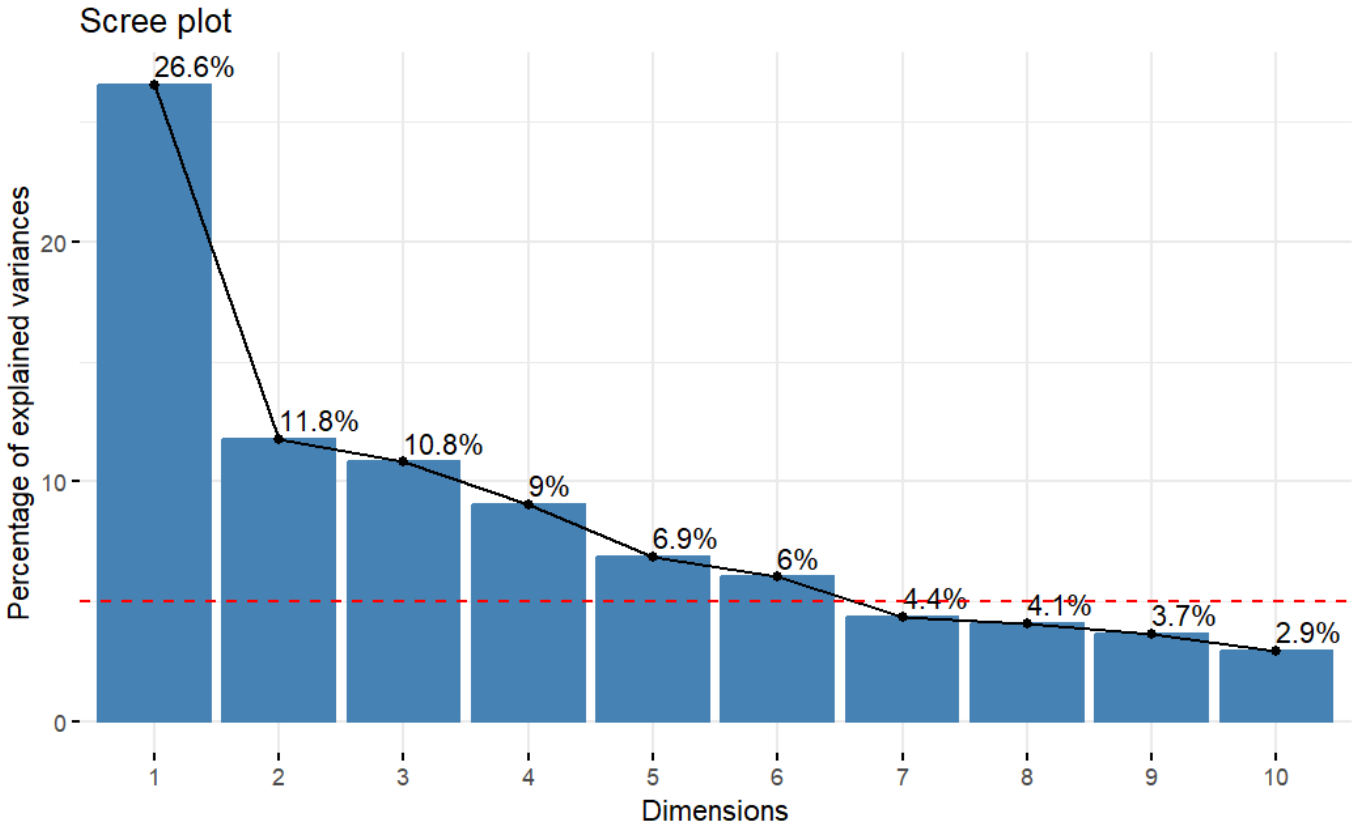
Puedes comentar el descarte de datos de las fuentes. Aquellos que no vayas a tener en cuenta por no ser útiles a tus objetivos.

Recomendamos utilizar RMarkDown ya que el análisis exploratorio se espera que lo hayáis hecho en R. Poned un anexo donde veamos las instrucciones, comandos, funciones, etc. además de las gráficas.

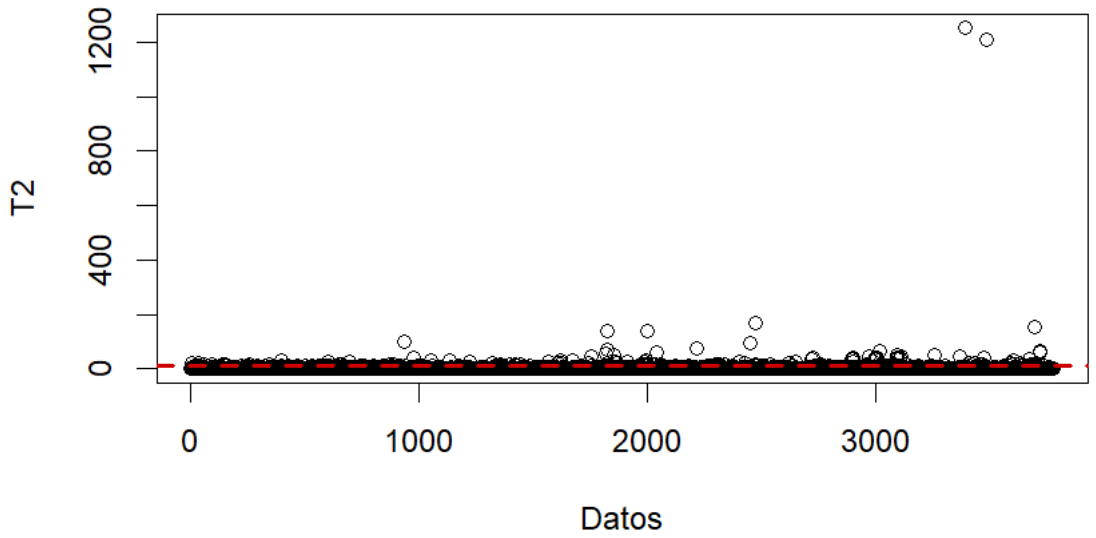
Para realizar el análisis exploratorio de datos de la base de datos original hemos comenzado revisando mediante la función `summary` de R los valores de los estadísticos más relevantes en cuanto a detección de anomalías (varianza mínima, valores atípicos, valores imposibles...). Se ha decidido no eliminar ninguna variable ya que todas presentan una leve variación.

Como segundo paso hemos visto las variables con valores faltantes. Decidimos eliminar las variables con un porcentaje de más de 25% de valores faltantes. Imputamos las variables de baño y de habitaciones con la mediana. También eliminamos las viviendas de las que no tenemos un valor de la reseña en las distintas categorías porque no podemos compararlas con el análisis de sentimiento. Finalmente, eliminamos las viviendas que no tienen valor del precio.

Después hemos revisado la cantidad de datos faltantes por variable y por observación y, por último, analizado valores anómalos mediante la aplicación de la T^2 de Hotteling. Antes de esto hemos realizado el PCA correspondiente seleccionando las 4 primeras componentes:



Los resultados fueron positivos, viendo pocos valores erróneos o sin sentido, alrededor de un 10% de valores faltantes en las que presentaban algunos (salvo un par de variables), y con pocos valores fuera de lo común según la T^2 de Hotteling.



Con estos resultados podemos concluir que la base de datos obtenida no brillaba por su calidad, pero sí nos es muy útil y nos deja un buen sabor de boca, pues además nos brindaba variables de especificación que, aunque no vamos a usar en nuestro análisis, pueden servir a futuros analistas que se interesen por el proyecto.

Con esto hecho partimos de una base robusta para el análisis de los datos y nos disponemos a conectarla con otra, ésta sobre renta media por distrito, extraída del Instituto Nacional de Estadística, con la que más adelante podremos llevar a cabo análisis y con la base de datos de las reseñas de la vivienda.

INE

En cuanto a la base de datos del INE, tenemos las variables: zona, renta media por persona, renta media por hogar, renta media por unidad de consumo, renta media bruta por persona y renta media bruta por hogar. Todas las variables numéricas contienen los datos desde el 2015 hasta el 2022.

La base de datos no estaba muy preparada para su uso en el análisis, entonces la hemos arreglado para ello. Después hemos hecho el análisis de valores faltantes que nos daba la siguiente tabla:

	Variable	numNA
:ona	Zona	0
:022_persona	2022_persona	25
:021_persona	2021_persona	30
:020_persona	2020_persona	41
:019_persona	2019_persona	35
:018_persona	2018_persona	35
:017_persona	2017_persona	31
:016_persona	2016_persona	31
:015_persona	2015_persona	31
:022_hogar	2022_hogar	25
:021_hogar	2021_hogar	30
:020_hogar	2020_hogar	41
:019_hogar	2019_hogar	35
:018_hogar	2018_hogar	35
:017_hogar	2017_hogar	31
:016_hogar	2016_hogar	31
:015_hogar	2015_hogar	31
:022_consumo	2022_consumo	25
:021_consumo	2021_consumo	30
:020_consumo	2020_consumo	41
:019_consumo	2019_consumo	35
:018_consumo	2018_consumo	35
:017_consumo	2017_consumo	34
:016_consumo	2016_consumo	34
:015_consumo	2015_consumo	34
:022_bruta_persona	2022_bruta_persona	25
:021_bruta_persona	2021_bruta_persona	30
:020_bruta_persona	2020_bruta_persona	41
:019_bruta_persona	2019_bruta_persona	35
:018_bruta_persona	2018_bruta_persona	35

Como casi todas las variables presentaban un número parecido de faltantes hemos decidido omitirlos (de 2468 observaciones a 2412).

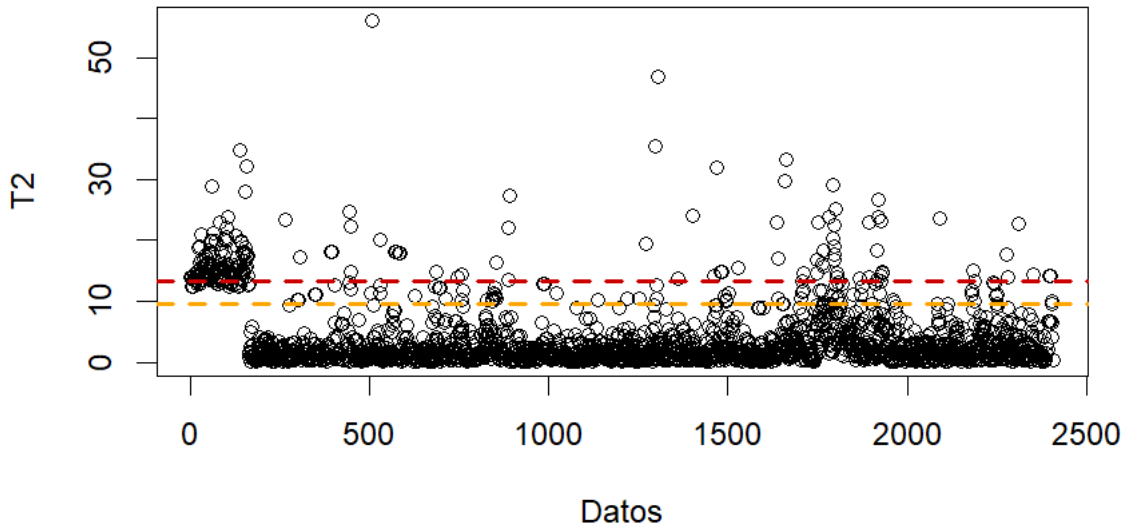
A continuación, realizamos el `summary` para observar los parámetros estadísticos correspondientes; y vamos a analizar el coeficiente de variación de las variables numéricas para deducir si alguna variable pudiera presentar *outliers*.

2021_bruta_persona	2022_bruta_persona	2021_persona	2020_bruta_persona
0.2213697	0.2218173	0.2226146	0.2249087
2022_persona	2019_bruta_persona	2020_persona	2018_bruta_persona
0.2281694	0.2291932	0.2330876	0.2345503
2019_persona	2017_bruta_persona	2021_consumo	2017_persona
0.2374973	0.2397920	0.2400540	0.2468373
2022_consumo	2016_bruta_persona	2020_consumo	2015_bruta_persona
0.2475007	0.2488317	0.2493741	0.2512845
2021_hogar	2019_consumo	2022_hogar	2018_persona
0.2527971	0.2541835	0.2567644	0.2587434
2016_persona	2015_persona	2020_hogar	2017_consumo
0.2589353	0.2599020	0.2620899	0.2621650
2019_hogar	2018_consumo	2015_consumo	2017_hogar
0.2632250	0.2723070	0.2734590	0.2739724
2016_consumo	2016_hogar	2015_hogar	2018_hogar
0.2747453	0.2818500	0.2824931	0.2849990
2021_bruta_hogar	2020_bruta_hogar	2022_bruta_hogar	2019_bruta_hogar
0.4235214	0.4297433	0.4297802	0.4369771
2017_bruta_hogar	2015_bruta_hogar	2018_bruta_hogar	2016_bruta_hogar
0.4425905	0.4474966	0.4478680	0.4568529

Podemos observar que las variables que presentan mayor variabilidad en este caso son las relacionadas con renta media bruta por hogar, por lo que son las que podrían presentar *outliers*.

Realizando los gráficos *boxplot* e histogramas correspondientes vamos que la gran mayoría de variables presentan valores atípicos por lo que realizamos una transformación logarítmica para intentar solucionarlo.

Tras la transformación logarítmica, la situación mejora un poco, pero se sigue encontrando una gran cantidad de atípicos:



ANEXO

Para iniciar el análisis, realizamos un ajuste en la base de datos eliminando aquellas variables que no vamos a utilizar en el estudio por diversos motivos, por ejemplo, porque la variable no varía o porque era irrelevante para el objetivo del análisis. Este proceso lo hemos realizado mediante la función `subset`.

```
# Limpiar listings.csv

```{r limpiar list}
# quitar las variables que no vamos a usar

list_limpiado = subset(val_list, select = -c(listing_url, scrape_id, last_scraped, source, picture_url, host_url, host_thumbnail_url,
host_picture_url, minimum_minimum_nights, maximum_minimum_nights, minimum_maximum_nights, maximum_maximum_nights,
minimum_nights_avg_ntm, maximum_nights_avg_ntm, calendar_updated, availability_30, availability_60, availability_90,
calendar_last_scraped, number_of_reviews_l30d, calculated_host_listings_count, room_type, bathrooms_text, beds,
calculated_host_listings_count, calculated_host_listings_count_entire_homes, calculated_host_listings_count_private_rooms,
calculated_host_listings_count_shared_rooms))
```

Además, hemos creado una nueva columna denominada `dollarPrice`, que representa la variable precio en formato numérico, eliminando el símbolo del dólar. Esta transformación permite trabajar con la variable de manera adecuada en los análisis y modelos predictivos posteriores.

```
# Crear una nueva columna para el precio sin el símbolo $
list_limpiado$dollarPrice <- as.numeric(gsub("\\$", "", list_limpiado$price))
```

De la misma manera, hemos considerado únicamente los pisos completos, excluyendo otros tipos de alojamiento, como habitaciones privadas. Esta elección se debe a que la dinámica de precios y las características de los alojamientos privados pueden diferir significativamente de las de los pisos enteros. Al enfocarnos únicamente en esta categoría, garantizamos una mayor homogeneidad en los datos, lo que contribuye a un análisis más preciso y representativo del mercado de alquiler de viviendas completas.

```
# Nos quedamos con los pisos enteros
list_limpiado_pisos = list_limpiado[grepl("entire", list_limpiado$property_type, ignore.case = TRUE), ]
```
```

Es necesario generar una tabla que contenga cada una de las variables junto con su tipo de dato (numérico, categórico, etc.). Para ello, utilizamos la función `data.frame`, permitiendo estructurar la información de manera clara y organizada.

La tabla resultante tiene el siguiente aspecto:

Description: df [49 × 2]

	variable <chr>	tipo <chr>
minimum_nights	minimum_nights	numerical
maximum_nights	maximum_nights	numerical
has_availability	has_availability	binary
availability_365	availability_365	numerical
number_of_reviews	number_of_reviews	numerical
number_of_reviews_ltm	number_of_reviews_ltm	numerical
first_review	first_review	text
last_review	last_review	text
review_scores_rating	review_scores_rating	numerical
review_scores_accuracy	review_scores_accuracy	numerical

31-40 of 49 rows Previous 1 2 3 4 5 Next

Una vez que estructuramos la base de datos, comenzamos con el análisis exploratorio de las variables. Como primer paso, analizamos la distribución tanto de las variables numéricas como de las binarias (mediante la función `apply`). Este proceso nos permite comprender mejor la estructura de los datos antes de tomar decisiones basadas en ellos.

```r
```{r variables numéricas}

# Distribución variables numéricas
numer = descValList$variable[descValList$tipo == "numerical"]
summary(list_limpiado_pisos[,numer])
```

```r
```{r variables binarias}

# Distribución de las variables binarias
apply(list_limpiado_pisos[,descValList$variable[descValList$tipo == "binary"]], 2, table, useNA = "i")
```

El siguiente paso por realizar ha sido eliminar algunas variables. En este caso, las variables eliminadas son las siguientes: "has_availability","property_type", "neighbourhood" y "host_neighbourhood".

En el caso de "has_availability", su problema residía en ser una variable binaria casi constante con 6311 observaciones True (1) y únicamente 27 false (0), lo cual apenas aporta información en un análisis a gran escala.

En cuanto a "property_type", la eliminamos porque filtramos por pisos enteros, no por tipo de propiedad.

"Neighbourhood" es eliminada debido a que sus observaciones son iguales en prácticamente todas, siendo estas "Valencian Community" (en distintos idiomas), ya sabiendo que claramente todos los Airbnb van a estar ubicados ahí.

Por último, "host_neighbourhood" nos resultaba una variable irrelevante para tener en cuenta, ya que el vecindario del 'host' no es algo significativo.

```
## Eliminamos las variables

```{r eliminar variables}

#quitamos las variables:

list_limpiado_pisos = list_limpiado_pisos[,setdiff(colnames(list_limpiado_pisos), c("has_availability",
"property_type", "neighbourhood", "host_neighbourhood"))]
descValList = descValList[colnames(list_limpiado_pisos),]
```
```

A continuación, hemos reemplazado los valores vacíos (" ") y los etiquetados como "NULL" por `NA` para poder trabajar con ellos, ya que representan datos faltantes. Utilizando la función `colSums(is.na())` calculamos la cantidad de valores ausentes en cada variable de tipo texto (`descValList$tipo="text"`).

```
```{r variables texto}

list_limpiado_pisos[list_limpiado_pisos == ""] = NA
list_limpiado_pisos[list_limpiado_pisos == "NULL"] = NA
valores_faltantes <- colSums(is.na(list_limpiado_pisos[, descValList$tipo == "text"]))
valores_faltantes
```
```

```
                          id                         name                  description
                           0                          0                          196
       neighborhood_overview                      host_id                    host_name
                        3154                          0                            0
                  host_since                host_location                   host_about
                           0                       1621                         2974
       host_response_time           host_response_rate      host_acceptance_rate
                           0                          0                            0
       host_neighbourhood           host_verifications              neighbourhood
                        4341                          0                         3154
 neighbourhood_cleansed neighbourhood_group_cleansed             property_type
                           0                          0                            0
                  amenities                      price                 first_review
                           0                        449                          974
                last_review                    license
                         974                       4250
```

También podemos obtener el porcentaje de valores faltantes para cada variable en la base de datos. Esto nos será más útil para identificar qué columnas tienen datos incompletos y tomar decisiones como imputar los datos faltantes.

```
# Valores faltantes o inconsistentes

```{r NAs, echo = TRUE}

numNA = apply(list_limpiado_pisos, 2, function(x) sum(is.na(x)))
percNA = round(100*apply(list_limpiado_pisos, 2, function(x) mean(is.na(x))), 2)
tablaNA = data.frame("Variable" = colnames(list_limpiado_pisos), numNA, percNA)
tablaNA
```

	Variable <chr>	num... <int>	percNA <dbl>
number_of_reviews	number_of_reviews	0	0.00
number_of_reviews_ltm	number_of_reviews_ltm	0	0.00
first_review	first_review	974	15.37
last_review	last_review	974	15.37
review_scores_rating	review_scores_rating	974	15.37
review_scores_accuracy	review_scores_accuracy	974	15.37
review_scores_cleanliness	review_scores_cleanliness	974	15.37
review_scores_checkin	review_scores_checkin	974	15.37
review_scores_communication	review_scores_communication	974	15.37
review_scores_location	review_scores_location	974	15.37

31-40 of 44 rows Previous 1 2 3 4 5 Next

Para refinar los datos, nos hemos fijado en aquellas variables con los valores faltantes más altos. Para "bathrooms" y "bedrooms" hemos imputado los valores faltantes con la mediana de todos aquellos con valor y hemos eliminado aquellas viviendas que carecían de precio y de valores en las reseñas, ya que son datos significativos e imputar no es óptimo.
```

```{r valores faltantes}
library(Hmisc)

# baños --> imputar con la mediana (1)
list_limpiado_pisos$bathrooms = impute(list_limpiado_pisos$bathrooms, fun = median)

# habitaciones --> imputar con la mediana (1)
list_limpiado_pisos$bedrooms <- impute(list_limpiado_pisos$bedrooms, fun = median)

#quitamos las viviendas que no tienen valores de las reseñas
list_limpiado_pisos <- list_limpiado_pisos[!is.na(list_limpiado_pisos$review_scores_rating), ]

#quitamos las viviendas que no tienen valores del precio
list_limpiado_pisos <- list_limpiado_pisos[!is.na(list_limpiado_pisos$price), ]

```

Y ahora observamos como bajan los valores faltantes en las variables transformadas:

```{r NAs depues, echo = TRUE}
numNA = apply(list_limpiado_pisos, 2, function(x) sum(is.na(x)))
percNA = round(100*apply(list_limpiado_pisos, 2, function(x) mean(is.na(x))), 2)
tablaNA = data.frame("Variable" = colnames(list_limpiado_pisos), numNA, percNA)
tablaNA
```

	Variable <chr>	num... <int>	percNA <dbl>
number_of_reviews	number_of_reviews	0	0.00
number_of_reviews_ltm	number_of_reviews_ltm	0	0.00
first_review	first_review	0	0.00
last_review	last_review	0	0.00
review_scores_rating	review_scores_rating	0	0.00
review_scores_accuracy	review_scores_accuracy	0	0.00
review_scores_cleanliness	review_scores_cleanliness	0	0.00
review_scores_checkin	review_scores_checkin	0	0.00
review_scores_communication	review_scores_communication	0	0.00
review_scores_location	review_scores_location	0	0.00

31-40 of 44 rows Previous 1 2 3 4 5 Next

Después de ver que hay casi la mitad de valores faltantes en "host_about" y "neighbourhood_overview", hemos comprobado por barrios la cantidad de viviendas y los faltantes en dicho barrio. El resultado ha sido observar cómo efectivamente de cada barrio faltan alrededor de la mitad de los valores.

	Variable <chr>	num... <int>	percNA <dbl>
id	id	0	0.00
name	name	0	0.00
description	description	138	2.74
neighborhood_overview	neighborhood_overview	2371	47.14
host_id	host_id	0	0.00
host_name	host_name	0	0.00
host_since	host_since	0	0.00
host_location	host_location	1310	26.04
host_about	host_about	2335	46.42
host_response_time	host_response_time	0	0.00

1-10 of 45 rows Previous 1 2 3 4 5 Next

```{r}

library(dplyr)

list_limpiado_pisos %>%
  group_by(neighbourhood_cleansed) %>%
  summarise(
    total_rows = n(),
    missing_neighborhood_overview = sum(is.na(neighborhood_overview))) %>%
  arrange(desc(missing_neighborhood_overview))
```

neighbourhood_cleansed <chr>	total_rows <int>	missing_neighborhood_overview <int>
CABANYAL-CANYAMELAR	594	239
RUSSAFA	397	146
AIORA	242	120
EL CARME	259	115
EL MERCAT	260	109
LA MALVA-ROSA	188	91
LA SEU	180	86
EL GRAU	146	81
MONT-OLIVET	168	79
EN CORTS	139	66

1-10 of 81 rows Previous 1 2 3 4 5 6 ... 9 Next

Para hacer un análisis sobre la predicción del precio de la vivienda, hemos decidido simplificar la base de datos eliminando todas aquellas observaciones con datos faltantes, tanto en la descripción del barrio, como en la información sobre el *host*. (Es importante recalcar que los datos sobre el *host* los reutilizaremos al completo más adelante para analizar la relación entre la fiabilidad del propietario y la disponibilidad de la vivienda. Sin embargo, hemos empleado esta limpieza para reducir la cantidad de valores faltantes en la base de datos).

```{r}
#quitamos las viviendas que no tienen valores del barrio ni sobre  el host
list_limpiado_pisos2 <- list_limpiado_pisos %>%
  filter(!(is.na(neighborhood_overview) & is.na(host_about)))
```

Volvemos a comprobar la cantidad de datos faltantes por variables y observamos como satisfactoriamente se han reducido casi a la mitad en las variables analizadas "host_about" y "neighbourhood_overview".

```{r}
numNA = apply(list_limpiado_pisos2, 2, function(x) sum(is.na(x)))
percNA = round(100*apply(list_limpiado_pisos2, 2, function(x) mean(is.na(x))), 2)
tablaNA = data.frame("Variable" = colnames(list_limpiado_pisos2), numNA, percNA)
tablaNA
```

	Variable <chr>	num... <int>	percNA <dbl>
id	id	0	0.00
name	name	0	0.00
description	description	122	3.23
neighborhood_overview	neighborhood_overview	1122	29.67
host_id	host_id	0	0.00
host_name	host_name	0	0.00
host_since	host_since	0	0.00
host_location	host_location	723	19.12
host_about	host_about	1086	28.72
host_response_time	host_response_time	0	0.00

1-10 of 45 rows Previous 1 2 3 4 5 Next

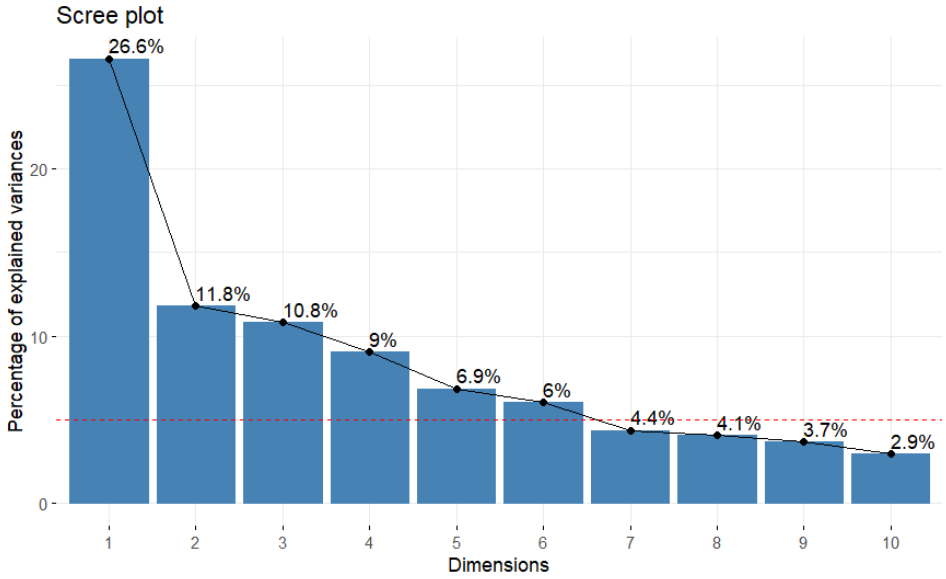
El último paso será analizar la variabilidad de las variables numéricas, calcular el coeficiente de variación y realizar el PCA. Para calcular la variabilidad de las variables numéricas sacamos la desviación típica de cada una ignorando los valores NA:

```
# Variabilidad de las variables numéricas (desviación típica)
```{r variabilidad}
mySD = apply(list_limpiado_pisos2[,descValList$variable[descValList$tipo == "numerical"]], 2, sd, na.rm=TRUE)
  #Calculamos la desviación típica (SD)
mySD
```
```

```
host_listings_count host_total_listings_count                 latitude
          105.61799500                204.19307736               0.02108297
             longitude                accommodates                 bathrooms
             0.02287217                  1.68952755               0.52619155
               bedrooms              minimum_nights             maximum_nights
             0.99418524                  8.02998349             443.89056891
        availability_365           number_of_reviews       number_of_reviews_ltm
            112.91682324                99.25173037              19.03432355
     review_scores_rating      review_scores_accuracy review_scores_cleanliness
             0.39060096                  0.34130669               0.36850346
review_scores_checkin review_scores_communication      review_scores_location
             0.31369815                  0.36585238               0.33938738
      review_scores_value           reviews_per_month
             0.40156651                  1.49690379
```

Sin embargo, es mejor calcular el coeficiente de variación (CV) ya que permite comparar la variabilidad entre diferentes variables sin importar sus unidades o magnitudes, esto es útil en análisis de precios para ver que variables son más dispersas.

```
# Mejor calcular el coeficiente de variación porque no depende de las unidades o magnitud de las variables
```{r coeficiente de variacion}
myMU = colMeans(list_limpiado_pisos2[,descValList$variable[descValList$tipo == "numerical"]], na.rm=TRUE)
myMU#La media
myCV = mySD/myMU  #Obtenemos el coeficiente de variación (que al dividir por la media eliminamos las magnitudes)
sort(myCV)  #Mostramos ordenando por coeficiente de variación
```
```

```
      host_listings_count   host_total_listings_count                    latitude
                29.682228                   39.562865                   39.466954
                longitude                 accommodates                   bathrooms
                -0.363334                    4.161538                    1.276127
                 bedrooms              minimum_nights              maximum_nights
                 1.698939                    4.916180                  516.320690
          availability_365           number_of_reviews       number_of_reviews_ltm
                200.229973                   74.080371                   18.879576
     review_scores_rating      review_scores_accuracy   review_scores_cleanliness
                 4.670732                    4.737419                    4.690032
    review_scores_checkin review_scores_communication      review_scores_location
                 4.800814                    4.798496                    4.696838
      review_scores_value           reviews_per_month
                 4.562775                    1.769618
                longitude                    latitude       review_scores_checkin
            -0.0629508183                0.0005341929                0.0653426950
   review_scores_accuracy      review_scores_location review_scores_communication
                0.0720448597                0.0722586904                0.0762431347
review_scores_cleanliness        review_scores_rating         review_scores_value
                0.0785716327                0.0836273529                0.0880092810
              accommodates                   bathrooms             availability_365
                0.4059862871                0.4123346763                0.5639356650
                 bedrooms           reviews_per_month              maximum_nights
                0.5851800696                0.8458908959                0.8597187326
    number_of_reviews_ltm           number_of_reviews              minimum_nights
                1.0081965799                1.3397844606                1.6333785353
      host_listings_count   host_total_listings_count
                3.5582906574                5.1612308360
```

El último paso realizado en este análisis exploratorio de las variables es el PCA.

```
##PCA
```{r pca}
library(FactoMineR)
library(factoextra)
res.pca = PCA(list_limpiado_pisos2, scale.unit = TRUE, graph = FALSE, ncp = 4, quali.sup = which(descValList$tipo
%in% c('text', 'binary')))
eig.val <- get_eigenvalue(res.pca)
VPmedio = 100 * (1/nrow(eig.val))
fviz_eig(res.pca, addlabels = TRUE) +
  geom_hline(yintercept=VPmedio, linetype=2, color="red")
K = 4
```
```

Con el método del codo se busca el punto donde la pendiente cambia drásticamente y la varianza explicada empieza a disminuir lentamente. En este caso, parece que después de la segunda o tercera componenta, la ganancia de información disminuye. Además, según la regla de Kaiser, si la línea roja representa el umbral del 5% de varianza explicada, los componentes por debajo pueden descartarse.

En base a la gráfica, probablemente las primeros 3 o 4 componentes sean suficientes para capturar la mayor parte de la información del conjunto de datos.

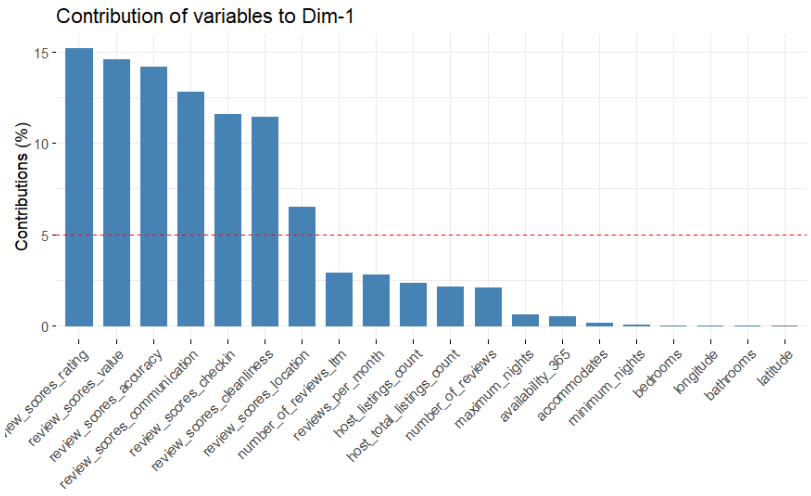
Nos quedaremos con 4 componentes de forma que más adelante podamos analizar 2 a 2 las componentes sin repetir ninguna.

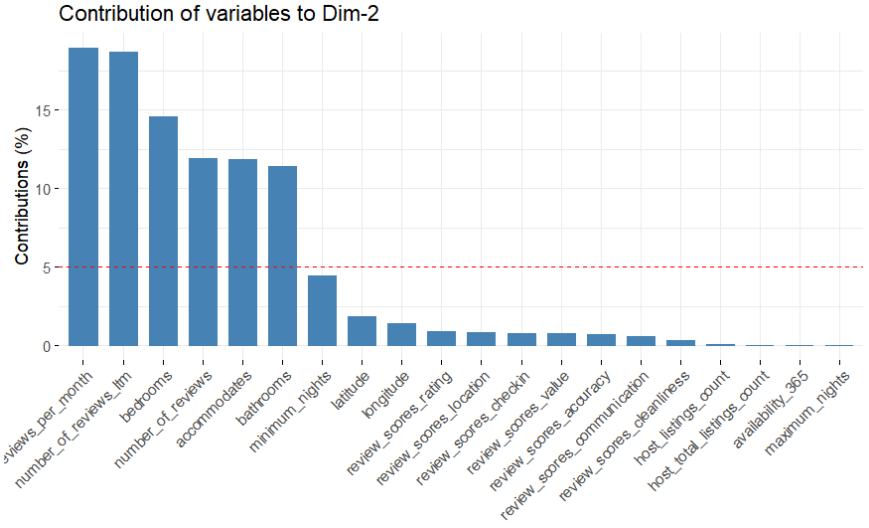


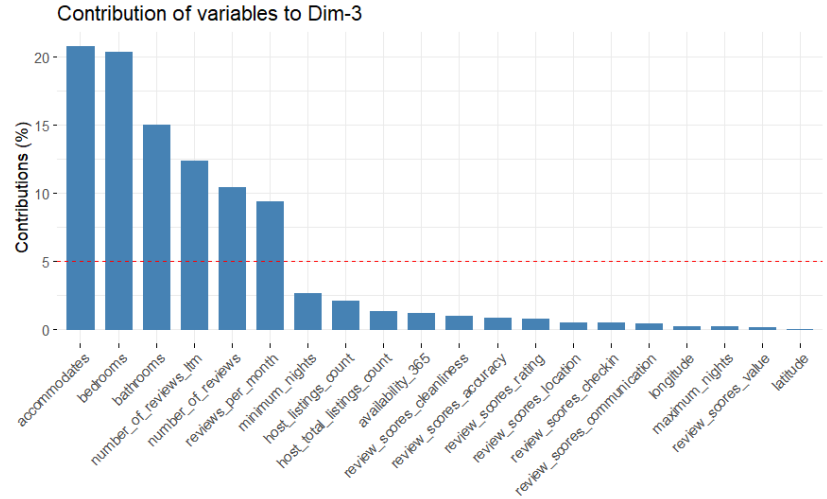

```{r pca dimensiones}
res.pca$var$contrib
fviz_contrib(res.pca, choice = "var", axes = 1) # PC1
fviz_contrib(res.pca, choice = "var", axes = 2) # PC2
fviz_contrib(res.pca, choice = "var", axes = 3) # PC3
fviz_contrib(res.pca, choice = "var", axes = 4) # PC4
```

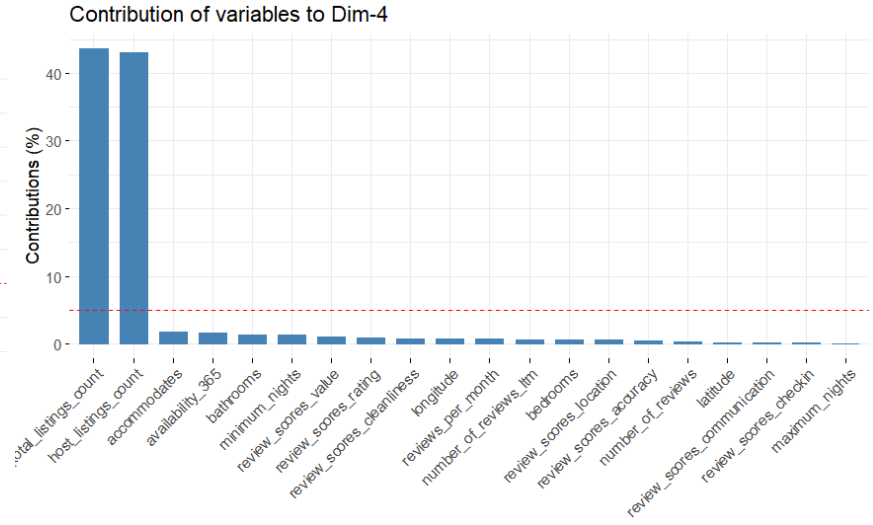
Calculamos la contribución de cada variable en las 4 primeras dimensiones, siendo las variables de *scores* las que más contribuyen a la primera dimensión y las de *reviews* en la segunda.

Además, la capacidad de alojamiento es la más contribuyente a la tercera dimensión (capacidad de personas, baños, camas) y listings a la cuarta dimensión.


Contribution of variables to Dim-1
Contributions (%)
0
5
10
15
review_scores_rating
review_scores_value
review_scores_accuracy
review_scores_communication
review_scores_checkin
review_scores_cleanliness
review_scores_location
number_of_reviews_ltm
reviews_per_month
host_listings_count
host_total_listings_count
number_of_reviews
maximum_nights
availability_365
accommodates
minimum_nights
bedrooms
longitude
bathrooms
latitude

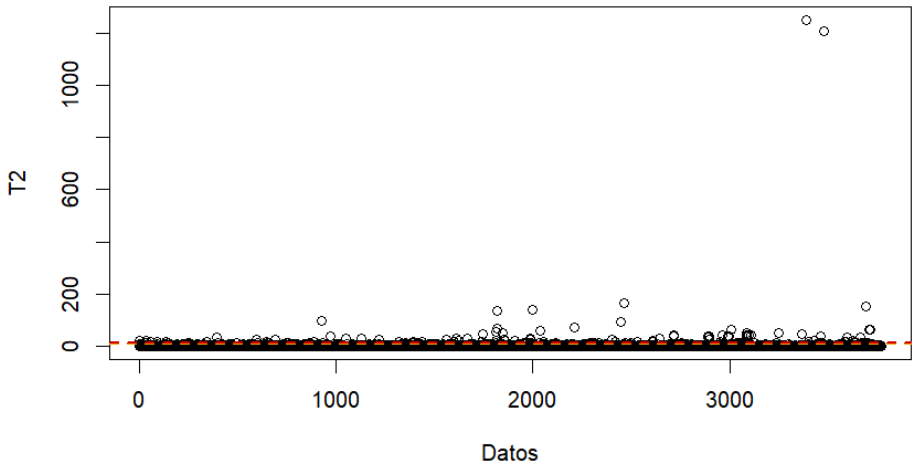

Contribution of variables to Dim-2
Contributions (%)
0
5
10
15
reviews_per_month
number_of_reviews_ltm
bedrooms
number_of_reviews
accommodates
bathrooms
minimum_nights
latitude
longitude
review_scores_rating
review_scores_location
review_scores_checkin
review_scores_value
review_scores_accuracy
review_scores_communication
review_scores_cleanliness
host_listings_count
host_total_listings_count
availability_365
maximum_nights


Contribution of variables to Dim-3
Contributions (%)
0
5
10
15
20
accommodates
bedrooms
bathrooms
number_of_reviews_ltm
number_of_reviews
reviews_per_month
minimum_nights
host_listings_count
host_total_listings_count
availability_365
review_scores_cleanliness
review_scores_accuracy
review_scores_rating
review_scores_location
review_scores_checkin
review_scores_communication
longitude
maximum_nights
review_scores_value
latitude


Contribution of variables to Dim-4
Contributions (%)
0
10
20
30
40
host_total_listings_count
host_listings_count
accommodates
availability_365
bathrooms
minimum_nights
review_scores_value
review_scores_rating
review_scores_cleanliness
longitude
reviews_per_month
number_of_reviews_ltm
bedrooms
review_scores_location
review_scores_accuracy
number_of_reviews
latitude
review_scores_communication
review_scores_checkin
maximum_nights

Más adelante, realizamos el T2 de Hotelling para observar los datos anómalos en las distintas observaciones:

```{r t de hotelling}
misScores = res.pca$ind$coord[,1:K]
miT2 = colSums(t(misScores**2)/eig.val[1:K,1])
I = nrow(list_limpiado_pisos2)
F95 = K*(I**2 - 1)/(I*(I - K)) * qf(0.95, K, I-K)
F99 = K*(I**2 - 1)/(I*(I - K)) * qf(0.99, K, I-K)

plot(1:length(miT2), miT2, type = "p", xlab = "Datos", ylab = "T2")
abline(h = F95, col = "orange", lty = 2, lwd = 2)
abline(h = F99, col = "red3", lty = 2, lwd = 2)
```



Guardamos en una variable todas las observaciones anómalas que superan el umbral F95, para así eliminarlas del análisis.

```{r quitar anómalas}
list_limpiado_pisos_sin_anom <- list_limpiado_pisos2[-anomalas, ]
```

```{r quitar anomalas}
list_limpiado_pisos_sin_anom <- list_limpiado_pisos2[-anomalas, ]
```

Por último, unimos el conjunto de datos de las viviendas limpio con el de las reseñas usando la columna `listing_id` como clave.

```
# Unir las reseñas con las viviendas

```{r unir}
library(dplyr)
colnames(list_limpiado_pisos_sin_anom)[colnames(list_limpiado_pisos_sin_anom) == "id"] <- "listing_id"
head(list_limpiado_pisos_sin_anom, 3)
unido = merge(list_limpiado_pisos_sin_anom, val_rev, by = "listing_id", all.x = TRUE)
head(unido, 3)
```
```