



```

1  ---
2  format: html
3  editor: visual
4  author: Juan David Pardo
5  ---
6
7  ---{r}
8  install.packages("ggplot2")
9  install.packages("dplyr")
10 ---
11
12 ---{r}
13 library(ggplot2)
14 library(dplyr, warn.conflicts = FALSE)
15 ---
16
17 Cargar los datos en un dataframe llamado: airbnb
18
19 ---{r}
20 airbnb<-read.csv('data/airbnb.csv',sep = ',', stringsAsFactors = T)
21
22 ##Llama nombre de las columnas
23 colnames(airbnb)
24
25 ##Llama la tabla airbnb
26 airbnb
27
28 ---

```






Description: df [13,207 x 13]

Zipcode	Neighbourhood.Cleansed	Property.Type	Room.Type	Accommodates	Bathrooms	Bedrooms	Beds	Bed.Type
28004	Universidad	Apartment	Private room	2	2.0	1	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	6	1.0	3	5	Real Bed
28004	Universidad	Apartment	Entire home/apt	3	1.0	2	2	Real Bed
28004	Universidad	Loft	Entire home/apt	3	2.0	1	1	Real Bed
28015	Universidad	Apartment	Entire home/apt	5	1.0	1	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	2	1.0	0	1	Real Bed
28004	Universidad	House	Private room	7	5.0	1	7	Real Bed
28004	Universidad	Apartment	Entire home/apt	3	1.0	1	NA	Real Bed
28015	Universidad	Apartment	Entire home/apt	2	1.0	1	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	4	2.0	2	2	Real Bed

1-10 of 13,207 rows | 1-9 of 13 columns

Previous 1 2 3 4 5 6 ... 100 Next

```

[1] "Zipcode"
"Room.Type"
[7] "Bedrooms"
"Amenities"
[13] "Review.Scores.Rating"

"Neighbourhood.Cleansed"
"Accommodates"
"Beds"
"Square.Feet"

"Property.Type"
"Bathrooms"
"Bed.Type"
"Price"

```

```

30 Mostrar las primeras 6 filas del dataframe
31
32 Renombrar las columnas de la siguiente forma:
33
34 | Nombre original | Nuevo nombre |
35 |-----|-----|
36 | Zipcode         | CodigoPostal |
37 | Neighbourhood.Cleansed | Barrio      |
38 | Property.Type    | TipoPropiedad |
39 | Room.Type        | TipoAlquiler |
40 | Accommodates     | MaxOcupantes |
41 | Bathrooms        | NumBanyos    |
42 | Bedrooms         | NumDormitorios |
43 | Beds            | NumCamas     |
44 | Bed.Type        | TipoCama     |
45 | Amenities       | Comodidades  |
46 | Square.Feet     | PiesCuadrados |
47 | Price          | Precio       |
48 | Review.Scores.Rating | Puntuacion  |
49
50 ---{r}
51 newnames<-c("CodigoPostal", "Barrio", "TipoPropiedad", "TipoAlquiler", "MaxOcupantes", "NumBanyos",
52 "NumDormitorios", "NumCamas", "TipoCama", "Comodidades", "PiesCuadrados", "Precio", "Puntuacion")
53 ---
54
55 ---{r}
56 ## Actualiza las nuevas cabeceras de la tabla airbnb
57 colnames(airbnb) <- newnames
58
59 ##Llama nombre de las columnas modificados
60 colnames(airbnb)
61
62 ##Valida las modificaciones de airbnb
63 airbnb
64 ---

```

R Console

data.frame
13,207 x 13

Description: df [13,207 x 13]

CodigoPostal	Barrio	TipoPropiedad	TipoAlquiler	MaxOcupantes	NumBanyos	NumDormitorios	NumCamas	TipoCama
28004	Universidad	Apartment	Private room	2	2.0	1	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	6	1.0	3	5	Real Bed
28004	Universidad	Apartment	Entire home/apt	3	1.0	2	2	Real Bed
28004	Universidad	Loft	Entire home/apt	3	2.0	1	1	Real Bed
28015	Universidad	Apartment	Entire home/apt	5	1.0	1	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	2	1.0	0	1	Real Bed
28004	Universidad	House	Private room	7	5.0	1	7	Real Bed
28004	Universidad	Apartment	Entire home/apt	3	1.0	1	NA	Real Bed
28015	Universidad	Apartment	Entire home/apt	2	1.0	1	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	4	2.0	2	2	Real Bed

1-10 of 13,207 rows | 1-9 of 13 columns

Previous 1 2 3 4 5 6 ... 100 Next

R Console

data.frame
13,207 x 13

```
[1] "CodigoPostal" "Barrio" "TipoPropiedad" "TipoAlquiler" "MaxOcupantes"
"NumBanyos" "NumDormitorios" "NumCamas" "TipoCama"
[10] "Comodidades" "PiesCuadrados" "Precio" "Puntuacion"
```

R Console

data.frame
13,207 x 13

```
1. Crea una nueva columna llamada MetrosCuadrados a partir de la columna PiesCuadrados.
Ayuda: 1 pie cuadrado son 0,092903 metros cuadrados
...{r}
conversion_factor <- 0.092903
airbnb <- airbnb %>% mutate(MetrosCuadrados = PiesCuadrados * conversion_factor)
airbnb_pies_metros <- airbnb[c("CodigoPostal", "Barrio", "TipoPropiedad", "TipoAlquiler", "PiesCuadrados", "MetrosCuadrados")]
airbnb_pies_metros
...
77
```

CodigoPostal	Barrio	TipoPropiedad	TipoAlquiler	PiesCuadrados	MetrosCuadrados
28004	Universidad	Apartment	Private room	NA	NA
28004	Universidad	Apartment	Entire home/apt	NA	NA
28004	Universidad	Apartment	Entire home/apt	NA	NA
28004	Universidad	Loft	Entire home/apt	NA	NA
28015	Universidad	Apartment	Entire home/apt	538	49.981814
28004	Universidad	Apartment	Entire home/apt	NA	NA
28004	Universidad	House	Private room	NA	NA
28004	Universidad	Apartment	Entire home/apt	NA	NA
28015	Universidad	Apartment	Entire home/apt	700	65.032100
28004	Universidad	Apartment	Entire home/apt	NA	NA

1-10 of 13,207 rows

Previous 1 2 3 4 5 6 ... 100 Next

R Console

data.frame
13,207 x 13

```
78
79 ...{r}
80 print(unique(airbnb$CodigoPostal))
81
```

[1] 28004	28015	28013	28005	28012	28014	28045	27013	28007	28028	28009	28001
[14] 28006	28010	28002	28034	28050	28008	28011	28049	28038	28053	28047	28025
[27] 28024	28016	28036	28046	28039	28020	28003	28029	28054	28041	28026	28058
[40] 28030	28017	28027	28043	28033	28055	28021	28032	28037	28022	28042	28094
[53] 28002\n28002	28035	28040	28031	28044	28105	28023	2805	28051	Madrid 28004	20126	28051\n28051
[66] 28048	28056	20013	27004	28060	-	28	2815	25008	2015	28052	2804

77 Levels: - 20013 20126 2015 25008 27004 27013 28 28001 280013 28002\n28002 28003 28004 28005 28006 28007 28008 28009 28010 28011 28012 28013 28014 28015 ... Madrid 28004

2. Miremos el código postal. Es una variable con entradas erróneas. Hay valores como '', '-' y '28' que deberían ser considerados como NA. Así mismo también debería ser NA todos los que no comiencen por 28, ya que estamos con códigos postales de Madrid

El código postal 28002, 28004 y 28051 tienen entradas repetidas. Por ejemplo las entradas 28002\n20882 deberían ir dentro de 28002


El código 2804 debería ser 28004, 2805 debería ser 28005 y 2815 junto con 2815 debería ser 28015

Limpia los datos de la columna Codigo Postal

```

93- '''{r}
94
95 airbnb <- airbnb %>%
96   mutate(
97     CodigoPostal = case_when(
98       CodigoPostal %in% c("", "-", "28") | !grepl("A28", CodigoPostal) ~ NA_character_,
99       TRUE ~ as.character(CodigoPostal)
100   )
101   ) %>%
102   mutate(
103     CodigoPostal = gsub("^n.*", "", CodigoPostal)
104   )
105
106 airbnb$CodigoPostal[airbnb$CodigoPostal == "2804"] <- "28004"
107 airbnb$CodigoPostal[airbnb$CodigoPostal == "2805"] <- "28005"
108 airbnb$CodigoPostal[airbnb$CodigoPostal == "2815"] <- "28015"
109 airbnb$CodigoPostal[airbnb$CodigoPostal == "280013"] <- "28013"
110 airbnb$CodigoPostal[airbnb$CodigoPostal == "Madrid.28004"] <- "28004"
111 airbnb$CodigoPostal[airbnb$CodigoPostal == "280013"] <- "28013"
112
113 ## Tabla principal modificada
114 airbnb
115
116 ## Validación de Códigos Postales
117 airbnb$CodigoPostal
118
119 ## Imprimir el número de NA
120 n_na <- sum(is.na(airbnb$CodigoPostal))
121 print(n_na)
122
123 '''

```




data.frame 13207 x 14

R Console

```

"28014" "28005" "28045" "28045" "28045" "28045" "28045" "28045"
"28045"
[841] "28045" "28045" "28045" "28028" "28028" "28028" "28028" "28028"
"28028" "28028" "28028" "28006" "28001" NA "28001" "28006" "28006" "28001"
"28001"
[861] "28006" NA "28006" "28002" "28002" "28002" "28016" "28016"
"28036" "28036" "28036" "28039" NA "28020" "28020" "28020" NA
"28020"
[881] "28010" "28003" "28010" "28035" "28035" "28035" "28008" "28008" "28008"
"28008" "28008" "28008" "28008" "28008" "28008" "28029" "28034"
"28029"
[901] "28034" "28034" "28008" "28008" "28008" "28034" "28034" "28040" "28038"
"28038" "28047" "28011" "28011" "28011" "28011" "28025" "28025" "28041" "28026"
"28026"
[921] NA "28053" "28038" "28018" "28018" "28030" "28030" "28030" "28030"
"28030" "28021" "28031" "28031" NA "28037" "28037" "28042" "28042" "28011"
"28011"
[941] "28011" "28011" "28047" "28024" "28019" "28019" NA "28019" "28019"
"28019" "28019" "28054" "28019" "28019" "28019" "28025" "28025" "28017" "28017"
"28017"
[961] "28027" "28027" "28027" "28017" NA "28027" "28043" "28043" "28033"
"28021" "28021" "28033" "28033" "28005" "28013" "28005" "28013" "28005"
"28013"
[981] "28013" "28013" "28005" "28005" "28013" "28005" "28012" "28005" "28005" NA
"28005" "28013" "28012" "28005" "28013" "28005" NA NA "28013"
"28013"
[ reached getOption("max.print") -- omitted 12207 entries ]
[1] 448

```



data.frame 13207 x 14

R Console

Description: df [13,207 x 14]

CodigoPostal	Barrio	TipoPropiedad	TipoAlquiler	MaxOcupantes	NumBanyos	NumDormitorios	NumCamas	TipoCama
28004	Universidad	Apartment	Private room	2	2.0	1	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	6	1.0	3	5	Real Bed
28004	Universidad	Apartment	Entire home/apt	3	1.0	2	2	Real Bed
28004	Universidad	Loft	Entire home/apt	3	2.0	1	1	Real Bed
28015	Universidad	Apartment	Entire home/apt	5	1.0	1	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	2	1.0	0	1	Real Bed
28004	Universidad	House	Private room	7	5.0	1	7	Real Bed
28004	Universidad	Apartment	Entire home/apt	3	1.0	1	N/A	Real Bed
28015	Universidad	Apartment	Entire home/apt	2	1.0	1	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	4	2.0	2	2	Real Bed

1-10 of 13,207 rows | 1-9 of 14 columns

Previous 1 2 3 4 5 6 ... 100 Next

```

124
125 -----
126
127 3. Una vez limpios los datos ¿Cuales son los códigos postales que tenemos?
128
129 '''{r}
130 cat("\n Los códigos postales que tenemos son:\n")
131 print(unique(airbnb$CodigoPostal))
132 '''

```

Los códigos postales que tenemos son:

```

[1] "28004" "28015" "28013" "28005" "28012" "28014" "28045" NA "28007" "28028" "28009" "28001" "28006" "28010" "28002" "28034" "28050" "28008" "28011" "28049" "28038"
[22] "28053" "28047" "28025" "28019" "28024" "28016" "28036" "28046" "28039" "28020" "28003" "28029" "28054" "28041" "28026" "28058" "28018" "28030" "28017" "28027" "28043"
[43] "28033" "28055" "28021" "28032" "28037" "28022" "28042" "28094" "28035" "28040" "28031" "28044" "28105" "28023" "28051" "28850" "28048" "28056" "28060" "28052"

```

```

133 -----
134
135
136 4. ¿Cuales son los 5 códigos postales con más entradas? ¿Y con menos? ¿Cuántas entradas tienen?
137
138 ```{r}
139 ## Contar la frecuencia de cada código postal & ordenar de mayor a menor frecuencia
140 codigo_postal_freq <- table(airbnb$CodigoPostal)
141
142 sorted_freq <- sort(codigo_postal_freq, decreasing = TRUE)
143
144 ## Obtener los 5 códigos postales con la mayor frecuencia
145 top_5_most_common <- head(sorted_freq, 5)
146
147 ## Obtener los 5 códigos postales con la menor frecuencia
148 top_5_least_common <- tail(sorted_freq, 5)
149
150 ## Imprimir resultados
151 cat("\n Los 5 códigos postales con más entradas:\n")
152 print(top_5_most_common)
153
154 cat("\n Los 5 códigos postales con menos entradas:\n")
155 print(top_5_least_common)
156
157 ```

```

Los 5 códigos postales con más entradas:

```

28012 28004 28005 28013 28014
2060 1795 1195 1020 630

```

Los 5 códigos postales con menos entradas:

```

28058 28060 28094 28105 28850
1      1      1      1      1

```

```

158 -----
159
160
161 5. ¿Cuales son los barrios que hay en el código postal 28012?
162
163 ```{r}
164
165 ## Filtrar los valores únicos por Barrio para el código postal 28012
166 barrios_28012 <- airbnb$Barrio[airbnb$CodigoPostal == "28012"]
167 unique_barrios_28012 <- unique(barrios_28012)
168
169 ## Imprimir el listado de Barrio para el código postal 28012
170 cat("\n Lista de barrios unicos con el codigo 28012:\n")
171 print(unique_barrios_28012)
172
173 ## Imprimir el número de NA
174 n_barrios <- length(unique_barrios_28012)
175 cat("\n Numero de barrios totales con el codigo 28012:\n")
176 print(n_barrios)
177
178 ```

```

Lista de barrios unicos con el codigo 28012:

```

[1] Sol      Acacias  <NA>      Palos de Moguer Embajadores Cortes    Palacio  Universidad Delicias  Arapiles
[11] Atocha   Goya
125 Levels: Abrantes Acacias Adelfas Aeropuerto Aguilas Alameda de Osuna Almagro Almenara Almendrales Aluche Ambroz Amposta Apostol Santiago Arapiles Aravaca ... Zofio

```

Numero de barrios totales con el codigo 28012:

```

[1] 12

```

```

179 -----
180
181
182 6. ¿Cuántas entradas hay en cada uno de esos barrios para el código postal 28012?
183
184 ```{r}
185 ## Filtrar el número de entradas en cada barrio
186 barrios_28012 <- airbnb$Barrio[airbnb$CodigoPostal == "28012"]
187 entradas_por_barrio <- table(barrios_28012)
188
189 ## Imprimir el resultado
190 cat("\n Numero de entradas por barrio para el codigo postal 28012: \n")
191 print(entradas_por_barrio[entradas_por_barrio != 0])
192
193 ```

```

Numero de entradas por barrio para el codigo postal 28012:

```

barrios_28012
Acacias      13
Arapiles      1
Atocha        1
Cortes       216
Delicias      1
Embajadores  1449
Goya          1
Palacio       27
Palos de Moguer 46
Sol          301
Universidad    4

```

```

194 -----
195
196
197 7. ¿Cuántos barrios hay en todo el dataset airbnb? ¿Cuales son?
198
199 ```{r}
200 # Obtener los valores únicos en la columna "Barrio"
201 barrios_unicos <- levels(airbnb$Barrio)
202
203 # Contar cuántos barrios únicos hay
204 num_barrios <- length(barrios_unicos)
205
206 # Imprimir el número de barrios y la lista de barrios
207 cat("Número total de barrios en el dataset de airbnb es:", num_barrios, "\n")
208 cat("Barrios:\n")
209 for(barrio in barrios_unicos){
210   cat(" *",barrio,"\n")
211 }
212
213 ```

```

Número total de barrios en el dataset de airbnb es: 125

Barrios:

- * Abrantes
- * Acacias
- * Adelfas
- * Aeropuerto
- * Aguilas
- * Alameda de Osuna
- * Almagro
- * Almenara
- * Almendrales
- * Aluche
- * Ambroz
- * Amposta
- * Apostol Santiago
- * Arapiles
- * Aravaca
- * Arcos
- * Arguelles

- * Atocha
- * Bellas Vistas
- * Berruquete
- * Buenavista
- * Butarque
- * Campamento
- * Canillas
- * Canillejas
- * Cármenes
- * Casa de Campo
- * Casco Histórico de Barajas
- * Casco Histórico de Vallecas
- * Casco Histórico de Vicálvaro
- * Castellana
- * Castilla
- * Castillejos
- * Chopera
- * Ciudad Jardín
- * Ciudad Universitaria
- * Colina
- * Comillas
- * Concepción
- * Conralesjos
- * Cortes
- * Costillares
- * Cuatro Caminos
- * Cuatro Vientos
- * Delicias
- * El Goloso
- * El Plantío
- * El Viso
- * Embajadores
- * Entrevías
- * Estrella
- * Fontarrón
- * Fuente del Berro
- * Fuentelareina
- * Gatañade
- * Goya
- * Guindalera
- * Hellín
- * Hispanoamérica

- * Ibiza
- * Imperial
- * Jerónimos
- * Justicia
- * La Paz
- * Legazpi
- * Lista
- * Los Angeles
- * Los Rosales
- * Lucero
- * Marroquina
- * Media Legua
- * Mirasierra
- * Moscardó
- * Niño Jesús
- * Nueva España
- * Numancia
- * Opañel
- * Orcasitas
- * Orcasur
- * Pacífico
- * Palacio
- * Palomas
- * Palomeras Bajas
- * Palomeras Sureste
- * Palos de Moguer
- * Pavones
- * Peñagrande
- * Pilar
- * Pinar del Rey
- * Piovera
- * Portazgo
- * Pradolongo
- * Prosperidad
- * Pueblo Nuevo
- * Puerta Bonita
- * Puerta del Ángel
- * Quintana
- * Recoletos
- * Rejas
- * Ríos Rosas

- * Rosas
- * Salvador
- * San Andrés
- * San Cristóbal
- * San Diego
- * San Fermín
- * San Isidro
- * San Juan Bautista
- * San Pascual
- * Santa Eugenia
- * Simancas
- * Sol
- * Timón
- * Trafalgar
- * Universidad
- * Valdeacederas
- * Valdehuentos
- * Valdemarín
- * Valdezarza
- * Vallehermoso
- * Valverde
- * Ventas
- * Vinateros
- * Vista Alegre
- * Zofío

```

214 -----
215
216
217 8. ¿Cuales son los 5 barrios que tienen mayor número entradas?
218
219 ```{r}
220
221 ### Contar la frecuencia de por barrio
222 barrios_freq <- table(airbnb$Barrio)
223
224 sorted_barrios_freq <- sort(barrios_freq, decreasing = TRUE)
225
226 ### Obtener los 5 barrios con la mayor frecuencia
227 top_5_barrios <- head(sorted_barrios_freq, 5)
228
229
230 ### Imprimir resultados
231 cat("\n los 5 barrios que tienen mayor número entradas:\n")
232 print(top_5_barrios)
233
234

```

los 5 barrios que tienen mayor número entradas:

Embajadores	Universidad	Palacio	sol	Justicia
1844	1358	1083	940	785

```

235 -----
236
237
238 9. ¿Cuántos Tipos de Alquiler diferentes hay? ¿Cuales son? ¿Cuántas entradas en el dataframe hay por cada tipo?
239
240 ```{r}
241 tipos_alquiler <- unique(airbnb$TipoAlquiler)
242
243 ### Imprimir el numero de tipos de alquiler
244 cat("Tipos de Alquiler Diferentes:", length(tipos_alquiler), "\n")
245
246 ### Contar las entradas en el dataframe de acuerdo con el tipo de alquiler
247 entradas_por_tipo <- table(airbnb$TipoAlquiler)
248 cat("\n Entradas por Tipo de Alquiler: \n")
249 print(entradas_por_tipo)
250
251

```

Tipos de Alquiler Diferentes: 3

Entradas por Tipo de Alquiler:

Entire home/apt	Private room	Shared room
7903	5113	191

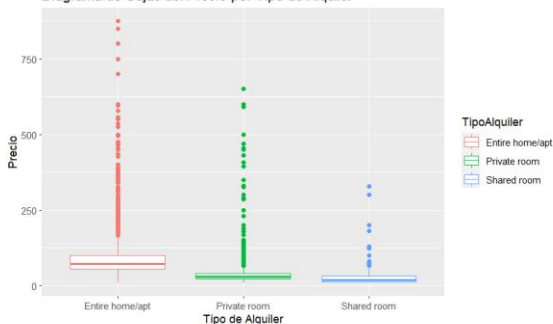
```

251 Muestra el diagrama de cajas del precio para cada uno de los diferentes Tipos de Alquiler
252
253 ```{r}
254
255 # Crear un diagrama de cajas del precio para cada tipo de alquiler
256 ggplot(data = airbnb, aes(x = TipoAlquiler, y = Precio)) +
257   geom_boxplot(aes(color=TipoAlquiler)) +
258   labs(title = "Diagrama de Cajas del Precio por Tipo de Alquiler", x = "Tipo de Alquiler", y = "Precio")
259
260
261

```

Warning: D[385:232mRemoved 9 rows containing non-finite values (stat_boxplot()) D[39m

Diagrama de Cajas del Precio por Tipo de Alquiler



```

262 -----
263
264
265 10. Cual es el precio medio de alquiler de cada uno, la diferencia que hay ¿es estadísticamente significativa? ¿Con que test lo comprobarías?
266
267 ```{r}
268
269 ### Calcular el precio medio para cada tipo de alquiler utilizando dplyr
270 precio_medio_por_tipo <- airbnb %>%
271   group_by(TipoAlquiler) %>%
272   summarize(PrecioMedio = mean(Precio, na.rm = TRUE))
273
274 ### Imprimir el resultado
275 print(precio_medio_por_tipo)
276
277
278 cat("\nKruskal-wallis test:\n",
279     "\n* H0 (hipótesis Nula): Las medianas de los grupos son iguales.\n",
280     "\n* H1 (hipótesis Alternativa): Al menos una mediana es diferente.\n",
281     "\nResultados del test:\n")
282
283 ### Realizar la prueba de Kruskal
284 kruskal_resultado <- kruskal.test(Precio ~ TipoAlquiler, data = airbnb)
285 ### Imprimir el resumen del kruskal
286 print(kruskal_resultado)
287
288 cat("El estadístico de Kruskal-Wallis es 7235.6 con 2 grados de libertad y un p-valor muy pequeño, menor que 2.2x10^-16. Existe evidencia significativa para rechazar la hipótesis nula, indicando que al menos una de las medianas de los grupos de tipo de alquiler es diferente.")
289
290 cat("\n Pruebas de wilcoxon para comparaciones pareadas:\n")
291
292 ### Realizar la prueba de comparacion wilcoxon emparejadas con ajuste de p-valor
293 resultado_wilcox <- pairwise.wilcox.test(airbnb$Precio, airbnb$TipoAlquiler, p.adjust.method = "holm")
294 # Imprimir el resultado
295 print(resultado_wilcox)
296
297 cat("\n El tamaño de los p-valor indica que existe evidencia significativa para rechazar la hipótesis nula en todas las comparaciones pareadas.\n",
298     "\n En conclusión podemos inferir que las comparaciones pareadas confirman que todas las parejas de grupos son significativamente diferentes en términos de precios. \n",
299     "\n Se hace una grafica de distribución de densidad por tipo de alquiler con el fin de validar la hipótesis H1. \n")
300

```

```

301 ##Grafica de distribución de densidad por tipo de alquiler
302 ggpplot(data=airbnb, aes(x=Precio,color=TipoAlquiler)) +
303   geom_density(bw=20) +
304   xlab('Precios') +
305   labs(title = "Distribución de densidad por tipo de alquiler",x = "Precio",y = "Densidad")
307

```



A tibble: 3 x 2

TipoAlquiler	PrecioMedio
Entire home/apt	87.29661
Private room	34.25514
Shared room	29.85340

308



data: Precio by TipoAlquiler

Kruskal-Wallis chi-squared = 7235.6, df = 2, p-value < 2.2e-16

El estadístico de Kruskal-Wallis es 7235.6 con 2 grados de libertad y un p-valor muy pequeño, menor que 2.2x10⁻¹⁶. Existe evidencia significativa para rechazar la hipótesis nula, indicando que al menos una de las medianas de los grupos de tipo de alquiler es diferente.

Pruebas de Wilcoxon para comparaciones pareadas:

Pairwise comparisons using Wilcoxon rank sum test with continuity correction

data: airbnb\$Precio and airbnb\$TipoAlquiler

```

      Entire home/apt Private room
Private room <2e-16      =
Shared room  <2e-16      <2e-16

```

P value adjustment method: holm

El tamaño de los p-valor indica que existe evidencia significativa para rechazar la hipótesis nula en todas las comparaciones pareadas.

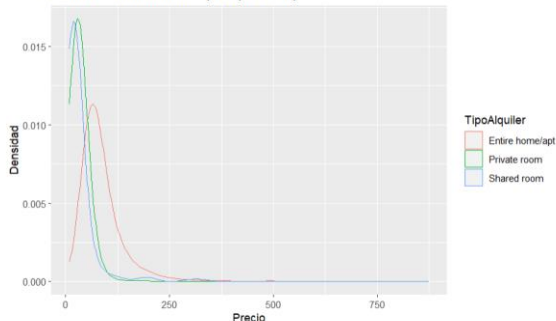
En conclusión podemos inferir que las comparaciones pareadas confirman que todas las parejas de grupos son significativamente diferentes en términos de precios.

Se hace una grafica de distribución de densidad por tipo de alquiler con el fin de validar la hipótesis H1.



Warning: [385:232mRemoved 9 rows containing non-finite values (stat_density()).]

Distribución de densidad por tipo de alquiler



311 11. Filtra el dataframe cuyos tipo de alquiler sea 'Entire home/apt' y guardalo en un dataframe llamado *airbnb_entire*. Estas serán las entradas que tienen un alquiler del piso completo.

```

312 ---{r}
313 ##Filtro de 'Entire home/apt'
314 airbnb_entire <- filter(airbnb, airbnb$TipoAlquiler == 'Entire home/apt')
315 # Visualizar las primeras filas del nuevo dataframe
316 print(airbnb_entire)
317
318 ...
319

```

Description: df [7,903 x 14]

CodigoPostal	Barrio	TipoPropiedad	TipoAlquiler	MaxOcupantes	NumBanyos	NumDormitorios	NumCamas	TipoCama
28004	Universidad	Apartment	Entire home/apt	6	1.0	3	5	Real Bed
28004	Universidad	Apartment	Entire home/apt	3	1.0	2	2	Real Bed
28004	Universidad	Loft	Entire home/apt	3	2.0	1	1	Real Bed
28015	Universidad	Apartment	Entire home/apt	5	1.0	1	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	2	1.0	0	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	3	1.0	1	NA	Real Bed
28015	Universidad	Apartment	Entire home/apt	2	1.0	1	1	Real Bed
28004	Universidad	Apartment	Entire home/apt	4	2.0	2	2	Real Bed
28004	Universidad	Apartment	Entire home/apt	4	1.0	1	2	Real Bed
28013	Sol	Apartment	Entire home/apt	4	1.0	1	1	Real Bed

1-10 of 7,903 rows | 1-9 of 14 columns

Previous 1 2 3 4 5 6 ... 100 Next

```

321 - -----
322
323 12. ¿Cuáles son los 5 barrios que tienen un mayor número de apartamentos enteros en alquiler? Nota: Mirar solo en airbnb_entire. A partir de este punto y hasta que se diga lo contrario partiremos de airbnb_entire.
324
325 ```{r}
326 # Top 5 de barrios con tipo de apartamento entero
327 top_barrios_entire <- head(sort(table(airbnb_entire$Barrio), decreasing = TRUE), 5)
328 # Imprimir los 5 barrios con el mayor número de apartamentos enteros
329 print(top_barrios_entire)
330
331 -

```

Embajadores	Universidad	Palacio	Sol	Cortes
1228	984	769	701	574

```

332 - -----
333
334 13. ¿Cuáles son los 5 barrios que tienen un mayor precio medio de alquiler para apartamentos enteros (es decir, del dataframe airbnb_entire)? ¿Cuál es su precio medio?
335
336 Ayuda: Usa la función aggregate `aggregate(.~colname,df,mean,na.rm=TRUE)`
337
338 ```{r}
339 precio_medio_por_barrio <- aggregate(Precio ~ Barrio, data = airbnb_entire, mean, na.rm = TRUE) %>% arrange(desc(Precio))
340 top_5_barrios_precio <- head(precio_medio_por_barrio, 5)
341 print(top_5_barrios_precio)
342
343 -

```

Description: df [5 x 2]

	Barrio	Precio
1	Palomas	309.7500
2	Fuente la Reina	180.0000
3	Recoletos	161.9254
4	El Plantío	150.0000
5	Castellana	141.3889

5 rows

```

346
347 14. ¿Cuántos apartamentos hay en cada uno de esos barrios?
348
349 Mostrar una dataframe con el nombre del barrio, el precio y el número de entradas.
350
351 Ayuda: Podéis crear un nuevo dataframe con las columnas "Barrio" y "Freq" que contenga el número de entradas en cada barrio y hacer un merge con el dataframe del punto anterior.
352
353 ```{r}
354 ## Cálculo del precio medio por barrio
355 precio_medio_por_barrio <- aggregate(Precio ~ Barrio, data = airbnb_entire, mean, na.rm = TRUE)
356
357 # Cálculo y Fusión de la tabla de frecuencia
358 df_freq <- data.frame(table(airbnb_entire$Barrio))
359 names(df_freq) <- c("Barrio", "Freq")
360 freq_precio_por_barrio <- merge(precio_medio_por_barrio, df_freq, by = "Barrio")
361
362 # Imprimir el resultado
363 print(freq_precio_por_barrio)
364
365
366 -

```

Description: df [119 x 3]

Barrio	Precio	Freq
Abrantes	46.00000	3
Acacias	68.16393	61
Adelfas	68.72727	33
Aeropuerto	38.00000	2
Aguilas	54.50000	2
Alameda de Osuna	138.75000	4
Almagro	109.18557	97
Almenara	65.68000	25
Almendrales	77.50000	18
Aluche	55.88889	9

1-10 of 119 rows

Previous 1 2 3 4 5 6 12 Next

```

369
370 15. Partiendo del dataframe anterior, muestra los 5 barrios con mayor precio, pero que tengan más de 100 entradas de alquiler.
371
372 ```{r}
373 mas_de_100 <- freq_precio_por_barrio %>% group_by(Barrio) %>% filter(Freq > 100)
374 top_5_mas_de_100 <- head(mas_de_100[order(mas_de_100$Precio, decreasing = TRUE)], 5)
375 print(top_5_mas_de_100)
376 -

```

A tibble: 5 x 3 Groups: Barrio [5]

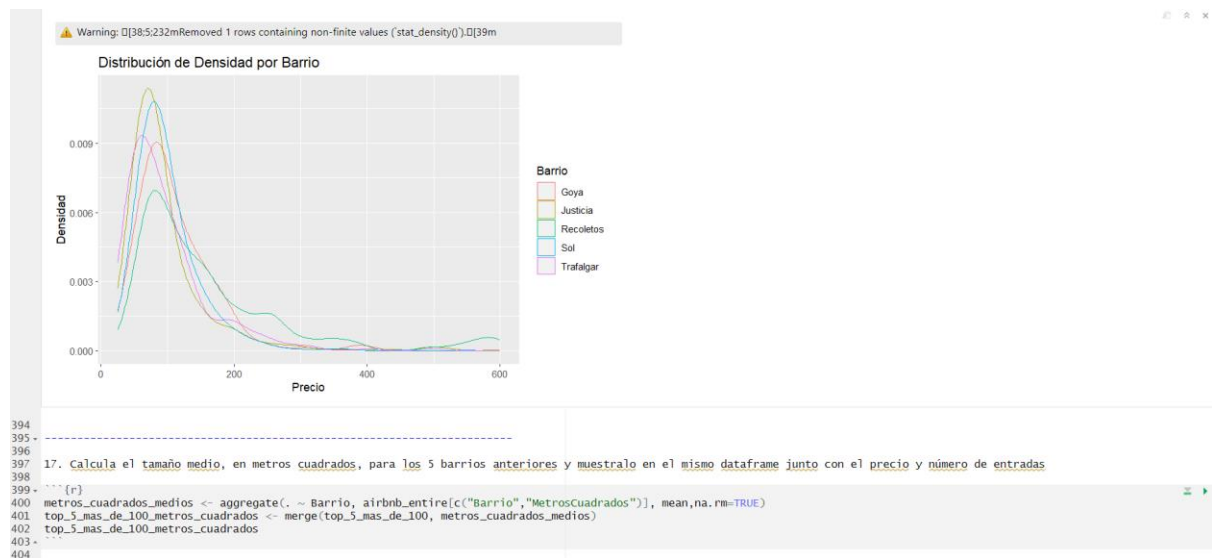
Barrio	Precio	Freq
Recoletos	161.92537	135
Goya	111.33803	142
Sol	100.75036	701
Trafalgar	98.57848	223
Justicia	98.25468	534

5 rows

```

380 16. Dibuja el diagrama de densidad de distribución de los diferentes precios. Serían 5 gráficas, una por cada barrio.
381
382 ```{r}
383
384 # Filtrar el dataframe por los top 5 barrios
385 top_barrios <- subset(airbnb_entire, Barrio %in% top_5_mas_de_100$Barrio)
386
387 # Graficar distribución de densidad por barrio
388 ggplot(top_barrios, aes(x = Precio, color = Barrio)) +
389   geom_density(bw = 20) +
390   xlab("Precios") +
391   labs(title = "Distribución de Densidad por Barrio", x = "Precio", y = "Densidad")
392
393 -

```

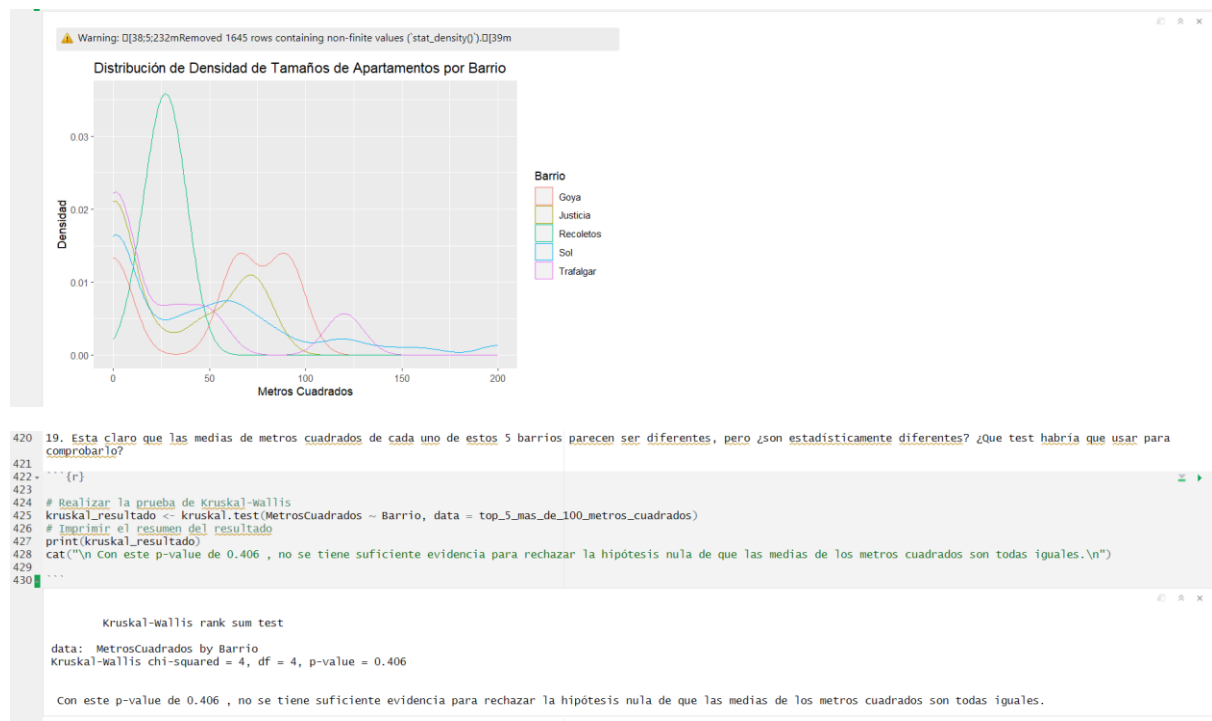



Description: df [5 x 4]

Barrio	Precio	Freq	MetrosCuadrados
Goya	111.33803	142	51.68504
Justicia	98.25468	534	28.52669
Recoletos	161.92537	135	26.66316
Sol	100.75036	701	45.61692
Trafalgar	98.57848	223	29.30426

5 rows

```
404
405
406
407 18. Dibuja el diagrama de densidad de distribución de los diferentes tamaños de apartamentos. Serían 5 gráficas, una por cada barrio.
408
409 '''{r}
410
411 ggplot(top_barrios,aes(x=MetrosCuadrados,color=Barrio)) +
412   geom_density(bw=10) +
413   labs(title = "Distribución de Densidad de Tamaños de Apartamentos por Barrio", x = "Metros Cuadrados", y = "Densidad")
414
415
416
```



```

431 -----
432
433
434 Para únicamente los pisos de alquiler en el barrio de Sol:
435
436 `barrio_sol<-subset(airbnb_entire,Barrio=="Sol")`
437
438 Calcular un modelo lineal que combine alguna de estas variables: \* NumBanyos \* NumDormitorios \* MaxOcupantes \* MetrosCuadrados
439
440 ```{r}
441 barrio_sol<-subset(airbnb_entire,Barrio=="Sol") %>% select(Barrio, NumBanyos, NumDormitorios, MaxOcupantes, MetrosCuadrados,Precio)
442 barrio_sol
443
444 ```

```

Description: df [701 x 6]

	Barrio	NumBanyos	NumDormitorios	MaxOcupantes	MetrosCuadrados	Precio
10	Sol	1.0	1	4	60.015338	75
11	Sol	1.0	2	6	NA	130
12	Sol	1.0	1	4	NA	100
13	Sol	1.0	2	6	NA	160
14	Sol	1.0	1	4	NA	49
15	Sol	1.0	1	4	NA	70
16	Sol	2.0	4	10	NA	210
17	Sol	1.0	2	6	NA	120
18	Sol	1.0	1	4	NA	50
19	Sol	1.0	1	4	69.955959	100

1-10 of 701 rows

Previous 1 2 3 4 5 6 ... 71 Next

```

446 -----
447 20. Primero calculamos la correlación para ver como se relacionan estas variables entre si.
448
449 ```{r}
450 correlacion_matriz <- cor(barrio_sol[, c("NumBanyos", "NumDormitorios", "MaxOcupantes", "MetrosCuadrados")],
451 use = "pairwise.complete.obs")
452 print(correlacion_matriz)
453
454 ```

```

```

      NumBanyos NumDormitorios MaxOcupantes MetrosCuadrados
NumBanyos      1.0000000      0.6761906      0.6578162      0.4820549
NumDormitorios  0.6761906      1.0000000      0.7594598      0.5684521
MaxOcupantes    0.6578162      0.7594598      1.0000000      0.4286233
MetrosCuadrados 0.4820549      0.5684521      0.4286233      1.0000000

```

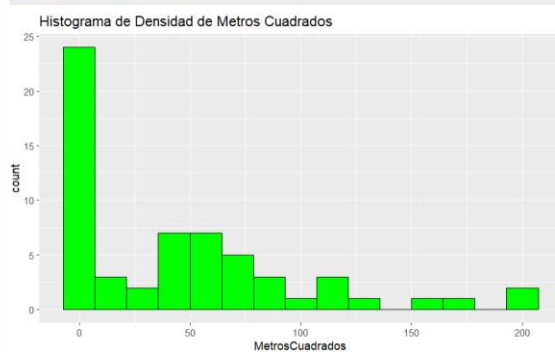
```

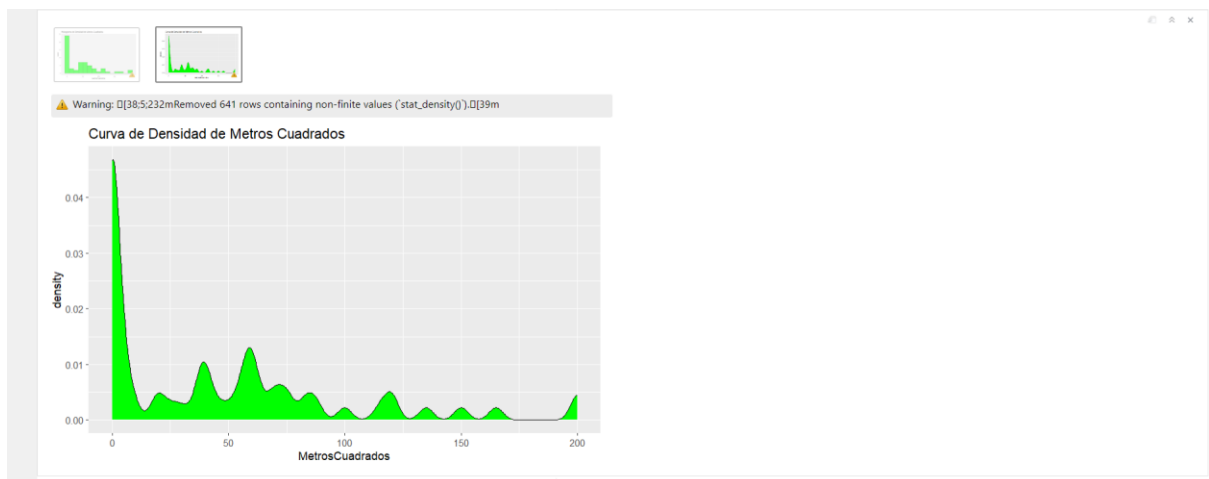
455 -----
456
457
458 21. Se observa que la correlación entre el número de dormitorios y los metros cuadrados es sorprendentemente baja. ¿Son de fiar esos números?
459
460 Mediante un histograma o curvas de densidad podemos descartar números que notienen sentido en el dataframe barrio_sol, para tener una matriz de correlación que tenga mayor sentido.
461
462 ```{r}
463 ### Histograma de densidad para los metros cuadrados
464 ggplot(data = barrio_sol, aes(x = MetrosCuadrados)) +
465   geom_histogram(bins=15, fill = "green", color = "black") +
466   labs(title = "Histograma de Densidad de Metros Cuadrados")
467
468 ## Curva de densidad para los metros cuadrados
469 ggplot(data = barrio_sol, aes(x = MetrosCuadrados)) +
470   geom_density(fill = "green", color = "black", alpha = 1, bw=3) +
471   labs(title = "Curva de Densidad de Metros Cuadrados")
472
473
474 ```

```

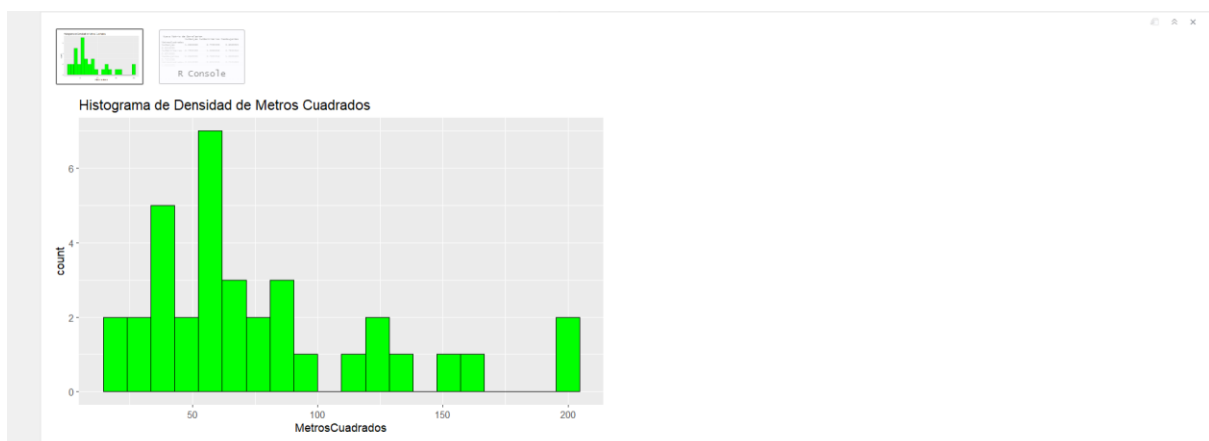


Warning: D[38:5232mRemoved 641 rows containing non-finite values (stat_bin).D[39m





```
475 -----
476 - '{r}'
477 -
478
479 barrio_sol_filtrado <- barrio_sol %>% filter(MetrosCuadrados > 15)
480
481
482 nueva_matriz_de_correlacion <- cor(barrio_sol_filtrado[, c("NumBanyos", "NumDormitorios", "MaxOcupantes", "MetrosCuadrados")],
483   use = "pairwise.complete.obs")
484
485
486
487 ### Histograma de densidad para los metros cuadrados
488 ggplot(data = barrio_sol_filtrado, aes(x = MetrosCuadrados)) +
489   geom_histogram(bins=20, fill = "red", color = "black") +
490   labs(title = "Histograma de Densidad de Metros Cuadrados")
491
492 cat("\n Nueva Matriz de Correlacion \n")
493 print(nueva_matriz_de_correlacion)
494
495 -
```



Nueva Matriz de Correlacion

	NumBanyos	NumDormitorios	MaxOcupantes	MetrosCuadrados
NumBanyos	1.0000000	0.7292599	0.8589205	0.6113095
NumDormitorios	0.7292599	1.0000000	0.7855310	0.8215065
MaxOcupantes	0.8589205	0.7855310	1.0000000	0.7121088
MetrosCuadrados	0.6113095	0.8215065	0.7121088	1.0000000

```

599 22. Una vez que hayamos filtrado los datos correspondientes calcular el valor o la combinación de valores que mejor nos permite obtener el precio de un inmueble.
600 ¿Que variable es más fiable para conocer el precio de un inmueble, el número de habitaciones o los metros cuadrados?
601
602 ```{r}
603
604 summary_modelo <- summary(lm(data = barrio_sol_filtrado, Precio ~ MetrosCuadrados + NumDormitorios + NumBanyos + MaxOcupantes))
605 summary_modelo
606 cat("\n La variable NumDormitorios resulta ser la mas fiable para conocer el precio del inmueble esto debido que tiene el P-valor muy pequeño\n")
607
608
609

```

```

Call:
lm(formula = Precio ~ MetrosCuadrados + NumDormitorios + NumBanyos +
    MaxOcupantes, data = barrio_sol_filtrado)

```

```

Residuals:
    Min       1Q   Median       3Q      Max
-76.852 -13.673  -5.048  18.523  69.473

```

```

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  28.0362    12.9584   2.164  0.03859 *
MetrosCuadrados  0.2237     0.2050   1.091  0.28393
NumDormitorios 28.2119     8.5440   3.302  0.00249 **
NumBanyos       0.5647     14.6038   0.039  0.96941
MaxOcupantes    2.6738     4.4329   0.603  0.55093
---

```

```

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

Residual standard error: 31.32 on 30 degrees of freedom
Multiple R-squared:  0.7588,    Adjusted R-squared:  0.7267
F-statistic: 23.6 on 4 and 30 DF,  p-value: 6.728e-09

```

La variable NumDormitorios resulta ser la mas fiable para conocer el precio del inmueble esto debido que tiene el P-valor muy pequeño

```

510 -----
511
512 23. Responde con su correspondiente margen de error del 95%, ¿cuantos euros incrementa el precio del alquiler por cada metro cuadrado extra del piso?
513
514 ```{r}
515
516 ##Nuevo modelo con margen de error del 95%
517 modelo_95 <- lm(Precio ~ MetrosCuadrados, data = barrio_sol_filtrado)
518 resumen_modelo_95 <- summary(modelo_95)
519
520 ## Obtener el intervalo de confianza del 95% para el coeficiente de MetrosCuadrados
521 intervalo_confianza <- confint(modelo_95, "MetrosCuadrados", level = 0.95)
522
523 ## Imprimir el intervalo de confianza
524 print(intervalo_confianza)
525 cat("\n Por cada metro cuadrado adicional, el precio de alquiler aumenta entre 0.604 y 1.185 euros.\n")
526
527
528
529

```

```

                2.5 %    97.5 %
MetrosCuadrados 0.6940201 1.26525

```

Por cada metro cuadrado adicional, el precio de alquiler aumenta entre 0.604 y 1.185 euros.

```

533 24. Responde con su correspondiente margen de error del 95%, ¿cuantos euros incrementa el precio del alquiler por cada habitación?
534
535 ```{r}
536
537 modelo3 <- lm(Precio ~ NumDormitorios, data = barrio_sol_filtrado)
538 # Imprimir el resumen del modelo
539 summary(model3)
540 # Obtener el intervalo de confianza del 95% para el coeficiente de NumDormitorios
541 confint(model3, level = 0.95)
542 cat("\n Por cada habitacion el precio de alquiler aumenta entre 39.484 con una variacion de mas o menos 31.03 euros.\n")
543
544

```

```

Call:
lm(formula = Precio ~ NumDormitorios, data = barrio_sol_filtrado)

```

```

Residuals:
    Min       1Q   Median       3Q      Max
-89.704 -17.010  -5.768  19.490  61.296

```

```

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  41.283      8.554   4.826 3.08e-05 ***
NumDormitorios 39.484      4.078   9.682 3.61e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

Residual standard error: 31.03 on 33 degrees of freedom
Multiple R-squared:  0.7396,    Adjusted R-squared:  0.7317
F-statistic: 93.75 on 1 and 33 DF,  p-value: 3.611e-11

```

```

                2.5 %    97.5 %
(Intercept)  23.87992 58.68671
NumDormitorios 31.18746 47.78095

```

Por cada habitacion el precio de alquiler aumenta entre 39.484 con una variacion de mas o menos 31.03 euros.

```

545 -----
546
547 25. ¿Cual es la probabilidad de encontrar, en el barrio de Sol, un apartamento en alquiler con 3 dormitorios? ¿Cual es el intervalo de confianza de esa probabilidad?
548
549
550
551 ```{r}
552 ##Conteo de inmuebles con 3 habitaciones en el barrio sol
553 barrio_sol_3_dormitorios <- sum(barrio_sol_filtrado$NumDormitorios == 3)
554 ##Conteo de inmuebles en el barrio sol
555 total_barrio_sol_3_dormitorios <- nrow(barrio_sol_filtrado)
556
557 ##Calculo de probabilidad
558 binom.test(barrio_sol_3_dormitorios, total_barrio_sol_3_dormitorios)
559 cat("\n* La probabilidad de alquilar un inmueble con 3 habitaciones en el barrio sol es del 8.57% \n",
560     "\n* Con un intervalo de confianza del 95% y un nuevo rango de probabilidad de exitoque se encuentra entre 1.803% a 2.305% \n")
561

```

Exact binomial test

```

data: barrio_sol_3_dormitorios and total_barrio_sol_3_dormitorios
number of successes = 3, number of trials = 35, p-value = 4.177e-07
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.01803764 0.23057502
sample estimates:
probability of success
 0.08571429

```

* La probabilidad de alquilar un inmueble con 3 habitaciones en el barrio sol es del 8.57%

* Con un intervalo de confianza del 95% y un nuevo rango de probabilidad de exitoque se encuentra entre 1.803% a 2.305%

