

Modulo 5: Técnicas Avanzadas de Predicción

Modelos Lineales Generalizados

Leandro Gutierrez

19/10/2024

Descripción de la tarea

Utilizando la base de datos de pisos, que hemos utilizado durante el temario en la que podemos encontrar un listado de pisos disponibles en Airbnb en Madrid, por temas computacionales, debes quedarte con un máximo de 2000 viviendas para responder las siguientes preguntas:

1. ¿Existe dependencia espacial en la variable precio? ¿Qué tipo de dependencia espacial existe: local, global o ambas?
2. Establece un modelo lineal para estimar la variable precio por m². ¿Hay dependencia espacial en los residuos del modelo?
3. Introduce una variable más en el modelo. Dicha variable es la distancia mínima entre cada persona y la geolocalización de las oficinas bancarias de Madrid obtenidas con OSM. ¿Sigue habiendo dependencia espacial en los residuos del nuevo modelo?
4. Modeliza el precio con un SAR. ¿Es significativo el factor de dependencia espacial? Interpreta el modelo.
5. Modeliza el precio con un SEM. ¿Es significativo el factor de dependencia espacial? Interpreta el modelo.
6. Valora la capacidad predictiva del modelo SAR con la técnica de validación cruzada.
7. Propón un modelo GWR para estimar los residuos con un cierto suavizado.

Solución

Carga de los datos

```
# cargamos el dataset
df <- read.csv('/Users/lgutierrez/Proyectos/master/M5/A3/data/table_5.05_2.csv')

df <- as_tibble(df)

summary(df)

##          X      longitude      latitude       price
##  Min.   : 1   Min.   :-3.836   Min.   :40.33   Min.   : 16.00
##  1st Qu.:1964  1st Qu.:-3.707   1st Qu.:40.41   1st Qu.: 56.00
##  Median :3941  Median :-3.702   Median :40.42   Median : 77.00
##  Mean   :3945  Mean   :-3.697   Mean   :40.42   Mean   : 97.57
##  3rd Qu.:5914  3rd Qu.:-3.695   3rd Qu.:40.43   3rd Qu.:110.00
##  Max.   :7905  Max.   :-3.567   Max.   :40.51   Max.   :999.00
##          room_type      minimum_nights   number_of_reviews review_scores_value
##  Length:7799      Min.   :1.000   Min.   : 1.00   Min.   : 2.000
##  Class :character  1st Qu.:1.000   1st Qu.: 6.00   1st Qu.: 9.000
```

```

##  Mode :character Median :2.000 Median : 23.00 Median : 9.000
##                Mean :2.272 Mean : 54.36 Mean : 9.175
##                3rd Qu.:3.000 3rd Qu.: 73.00 3rd Qu.:10.000
##                Max. :9.000 Max. :643.00 Max. :10.000
## calculated_host_listings_count    bedrooms reviews_per_month
## Min.   : 1.00      Min.   :1.000  Min.   : 0.010
## 1st Qu.: 1.00      1st Qu.:1.000  1st Qu.: 0.360
## Median : 2.00      Median :1.000  Median : 1.040
## Mean   :14.34      Mean   :1.693  Mean   : 1.559
## 3rd Qu.: 9.00      3rd Qu.:2.000  3rd Qu.: 2.300
## Max.  :213.00      Max.  :9.000   Max.  :11.130
##      beds accommodates availability_30 availability_60
## Min.   : 0.000  Min.   :1.000  Min.   : 0.00  Min.   : 0.00
## 1st Qu.: 2.000  1st Qu.: 3.000  1st Qu.: 0.00  1st Qu.: 0.00
## Median : 2.000  Median : 4.000  Median :14.00  Median :39.00
## Mean   : 2.553  Mean   : 4.402  Mean   :13.92  Mean   :30.59
## 3rd Qu.: 3.000  3rd Qu.: 6.000  3rd Qu.:29.00  3rd Qu.:58.00
## Max.  :17.000  Max.  :16.000  Max.  :30.00  Max.  :60.00
## availability_90 instant_bookable Distancia_Centro Distancia_Norte
## Min.   : 0.00  Length:7799  Min.   : 0.0  Min.   : 0.100
## 1st Qu.: 0.00  Class :character 1st Qu.: 0.7  1st Qu.: 5.400
## Median :64.00  Mode  :character  Median : 1.0  Median : 6.400
## Mean   :48.07
## 3rd Qu.:88.00
## Max.  :90.00
##      Distancia_Sur logprice tv_ports phone_ports
## Min.   : 0.10  Min.   :2.773  Min.   :1.000  Min.   :1.000
## 1st Qu.: 3.00  1st Qu.:4.025  1st Qu.:1.000  1st Qu.:2.000
## Median : 3.70  Median :4.344  Median :2.000  Median :3.000
## Mean   : 4.31  Mean   :4.406  Mean   :2.474  Mean   :2.504
## 3rd Qu.: 4.80  3rd Qu.:4.700  3rd Qu.:3.000  3rd Qu.:3.000
## Max.  :15.70  Max.  :6.907  Max.  :4.000  Max.  :4.000
##      Vecinos Piso ventanas
## Min.   :1.000  Min.   :1.000  Min.   : 1.000
## 1st Qu.:1.000  1st Qu.:2.000  1st Qu.: 2.000
## Median :2.000  Median :4.000  Median : 3.000
## Mean   :1.992  Mean   :3.516  Mean   : 3.177
## 3rd Qu.:3.000  3rd Qu.:5.000  3rd Qu.: 4.000
## Max.  :3.000  Max.  :6.000  Max.  :11.000

glimpse(df)

## #> Rows: 7,799
## #> Columns: 26
## #> $ X <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ~
## #> $ longitude <dbl> -3.69764, -3.70346, -3.70269, -3.69730, ~
## #> $ latitude <dbl> 40.41995, 40.41552, 40.41111, 40.41978, ~
## #> $ price <dbl> 115, 65, 54, 71, 90, 55, 79, 115, 80, 6~
## #> $ room_type <chr> "Entire home/apt", "Entire home/apt", "~"
## #> $ minimum_nights <int> 3, 5, 3, 3, 5, 5, 2, 3, 8, 3, 3, 1, 2, ~
## #> $ number_of_reviews <int> 68, 170, 8, 120, 48, 50, 110, 18, 5, 56~
## #> $ review_scores_value <int> 10, 10, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9~
## #> $ calculated_host_listings_count <int> 1, 3, 1, 15, 3, 1, 5, 8, 1, 1, 1, 1, 5, ~
## #> $ bedrooms <int> 2, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 3, ~
## #> $ reviews_per_month <dbl> 0.60, 1.37, 0.12, 1.84, 0.40, 1.64, 0.9~
```

```

## $ beds <int> 3, 2, 1, 2, 3, 1, 6, 1, 2, 2, 1, 1, 3, ~
## $ accommodates <int> 4, 2, 2, 4, 3, 2, 5, 4, 4, 3, 2, 2, 5, ~
## $ availability_30 <int> 14, 30, 0, 0, 30, 0, 18, 30, 28, 0, 0, ~
## $ availability_60 <int> 44, 60, 0, 0, 60, 0, 48, 60, 58, 0, 0, ~
## $ availability_90 <int> 74, 90, 0, 0, 90, 0, 78, 90, 88, 0, 0, ~
## $ instant_bookable <chr> "f", "f", "f", "t", "f", "f", "t", ~
## $ Distancia_Centro <dbl> 0.6, 0.1, 0.6, 0.6, 0.5, 0.9, 0.6, 1.0, ~
## $ Distancia_Norte <dbl> 5.9, 6.5, 7.0, 5.9, 6.0, 5.8, 7.0, 6.8, ~
## $ Distancia_Sur <dbl> 3.9, 3.3, 2.9, 3.9, 3.7, 3.9, 2.8, 3.4, ~
## $ logprice <dbl> 4.744932, 4.174387, 3.988984, 4.262680, ~
## $ tv_ports <int> 4, 3, 4, 2, 1, 3, 1, 2, 3, 2, 2, 4, 3, ~
## $ phone_ports <int> 2, 3, 4, 2, 4, 1, 3, 3, 2, 3, 2, 2, 1, ~
## $ Vecinos <int> 3, 2, 2, 3, 2, 3, 2, 2, 2, 2, 2, 2, ~
## $ Piso <int> 5, 6, 2, 5, 5, 1, 6, 2, 5, 5, 4, 6, 4, ~
## $ ventanas <int> 2, 1, 2, 3, 1, 2, 6, 3, 4, 3, 2, 1, 4, ~

dim(df)

## [1] 7799   26

```

Podemos observar que contamos con un dataset de **7799 observaciones**, con **26 variables**. No se observan valores nulos.

Apartado 1

Primero haremos un pequeño saneamiento de nuestro dataset para poder obtener la matriz de vecinos espaciales, eliminando coordenadas identicas y así poder determinar si existe correlación espacial en la variable logprice

```

# checkeamos si existen coordenadas duplicadas en el dataframe
duplicados <- df %>%
  group_by(longitude, latitude) %>%
  filter(n() > 1)

# quitamos coordenadas duplicadas de dataframe
df_nd <- df %>% distinct(longitude, latitude, .keep_all = TRUE)

# creamos la matriz de vecinos espaciales
nb <- knn2nb(knearneigh(cbind(df_nd$longitude, df_nd$latitude), k=5))

```

Ahora con la matriz de vecinos realizaremos el analisis de los test I-Moran y LISA

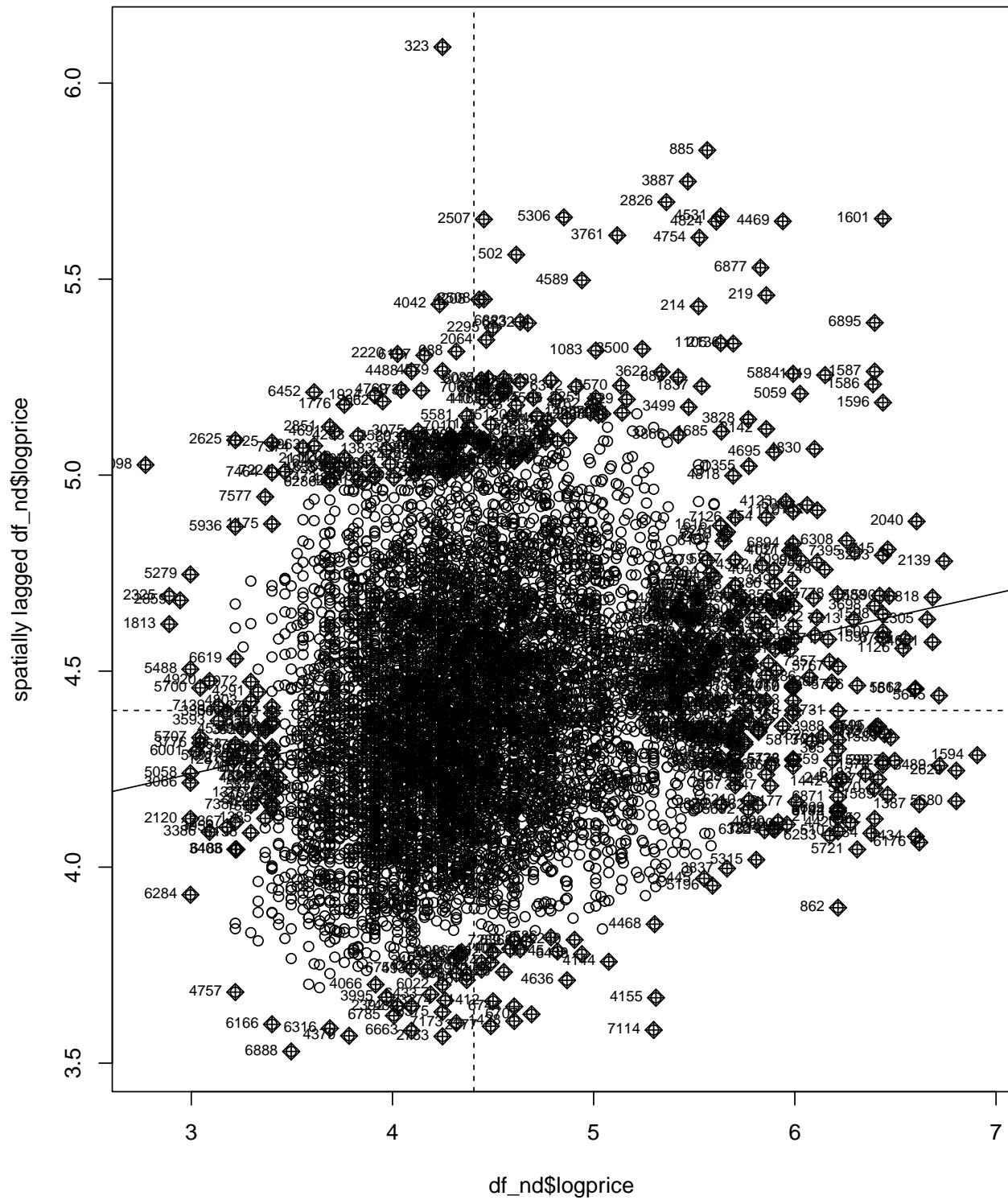
```
# obtenemos el test I-Moran
```

```
moran.test(x = df_nd$logprice, listw = nb2listw(nb, style="W"))
```

```
##
## Moran I test under randomisation
##
## data: df_nd$logprice
## weights: nb2listw(nb, style = "W")
##
## Moran I statistic standard deviate = 16.889, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##          1.150742e-01 -1.312680e-04    4.653034e-05
```

```
# visualizamos resultados
moran.plot(x = df_nd$logprice, listw = nb2listw(nb, style="W"), main="Gráfico I Moran")
```

Gráfico I Moran



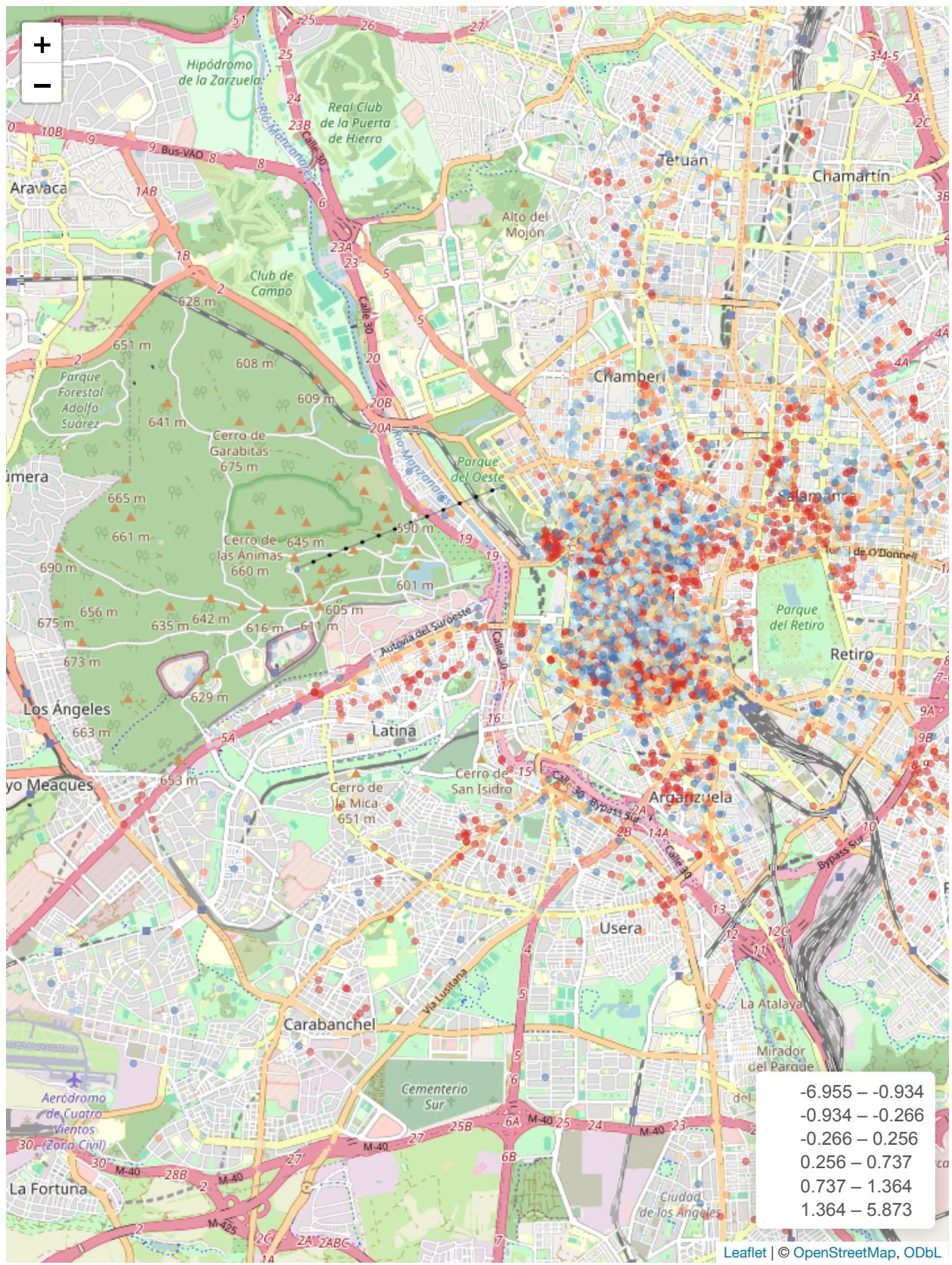
Vemos que el estadístico **I-Moran** que obtenemos tiene un valor de **1.150742e-01** con un p-value de **2.2e-16**.

Lo que nos indica que hay una leve correlación espacial positiva, altamente significativa.

Relizamos el test LISA para determinar si existe autocorrelación espacial local

```
# obtenemos el test LISA
local_moran <- as.data.frame(localmoran(x = df_nd$logprice, listw = nb2listw(nb, style="W")))

# visualizamos en el mapa
pl_pt(df_nd, color2 = local_moran$Z.Ii, size2=0.1, dd = 6)
```



Del análisis visual podemos ver apreciar ciertas zonas de sobrecarga de puntos rojos lo que nos podría estar

dando indicios de la existencia de dependencia espacial local, sobre todo en zona noroeste lindero al Parque del Retiro.

Apartado 2

Vamos a crear un modelo lineal como base para el desarrollo del apartado, durante el desarrollo se realizaron pruebas de StepAIC para determinar las variables a incluir en el modelo y no se llegó a registrar grandes mejoras respecto al modelo simplificado, por practicidad se opta por un modelo lineal con formula simple donde intervienen la mayoría de las variables

```
# convertimos las variables character a factores
df_nd <- df_nd %>%
  mutate_if(is.character, as.factor)

# quitamos las variables tipo factor que tienen menos de dos nivel
df_nd <- df_nd[, sapply(df_nd, function(x) !(is.factor(x) && nlevels(x) < 2))]

# creamos modelo completo con todas las variables excepto las variables room_type, price y X
modelo_1 <- lm(logprice ~ . - price, data=df_nd)

# visualizamos resumen del modelo
pander(summary(modelo_1))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-40,87	42,25	-0,9673	0,3334
longitude	-1,246	0,4549	-2,738	0,006198
latitude	0,9805	1,069	0,917	0,3591
minimum_nights	-0,02107	0,003847	-5,477	4,474e-08
number_of_reviews	0,0006148	0,0001152	5,337	9,712e-08
review_scores_value	0,03824	0,005407	7,073	1,657e-12
calculated_host_listings_count	-0,001294	0,0001513	-8,552	1,443e-17
bedrooms	0,1734	0,01066	16,26	1,795e-58
reviews_per_month	-0,1316	0,005771	-22,81	1,823e-111
beds	0,0157	0,005816	2,699	0,006962
accommodates	0,05236	0,004839	10,82	4,301e-27
availability_30	0,008015	0,001738	4,613	4,039e-06
availability_60	-0,004568	0,001975	-2,313	0,02074
availability_90	0,0004922	0,0009217	0,534	0,5934
instant_bookable	0,02778	0,01132	2,455	0,01413
Distancia_Centro	-0,1064	0,006864	-15,51	2,041e-53
Distancia_Norte	0,02314	0,007664	3,019	0,002548
Distancia_Sur	0,08952	0,01133	7,899	3,206e-15
tv_ports	0,005298	0,004472	1,185	0,2362
phone_ports	0,003504	0,004474	0,7833	0,4334
Vecinos	0,008268	0,006139	1,347	0,1781
Piso	-0,001175	0,002934	-0,4005	0,6888
ventanas	-0,00416	0,004508	-0,9227	0,3562

Table 2: Fitting linear model: logprice ~ . - X - price

Observations	Residual Std. Error	R ²	Adjusted R ²
7.619	0,4361	0,3558	0,3539

```
# encontramos SCR  
sum((modelo_1$resid)**2)
```

```
## [1] 1444.464
```

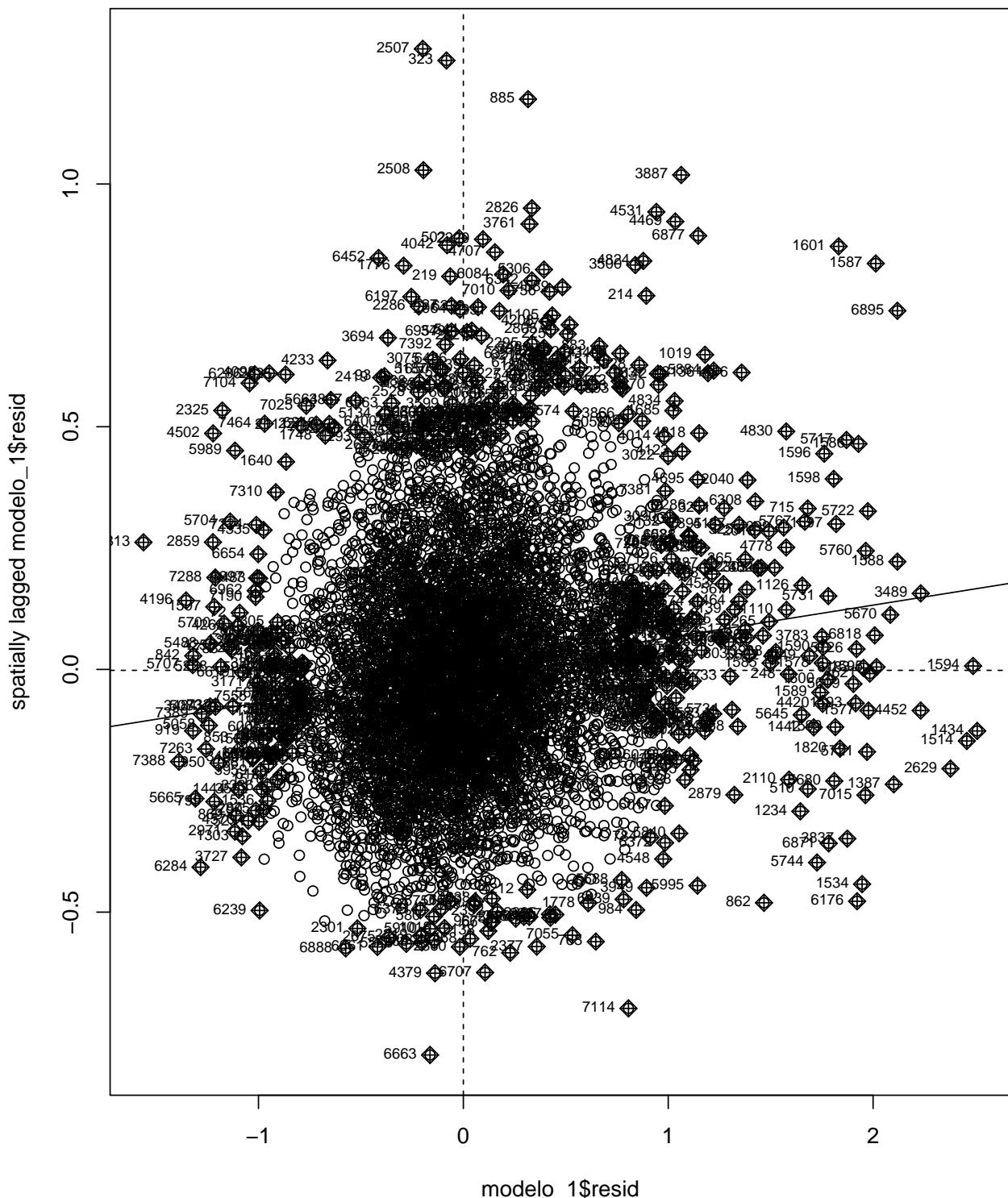
Obtenemos un modelo con un R_a^2 con un valor de **0,3509**, lo que nos dice que nuestras variables predictoras explican el 35% de la variabilidad total de nuestra incognita (price). Además obtenemos una *Suma de los Errores al Cuadrado* de **1444**.

Ahora con nuestro modelo listo vemos si existe dependencia espacial global

```
moran.test(x = modelo_1$resid, listw = nb2listw(nb, style="W"))
```

```
##  
## Moran I test under randomisation  
##  
## data: modelo_1$resid  
## weights: nb2listw(nb, style = "W")  
##  
## Moran I statistic standard deviate = 9.8705, p-value < 2.2e-16  
## alternative hypothesis: greater  
## sample estimates:  
## Moran I statistic      Expectation      Variance  
##       6.719334e-02     -1.312680e-04     4.652277e-05  
moran.plot(x = modelo_1$resid, listw = nb2listw(nb, style="W"), main="Gráfico I Moran")
```

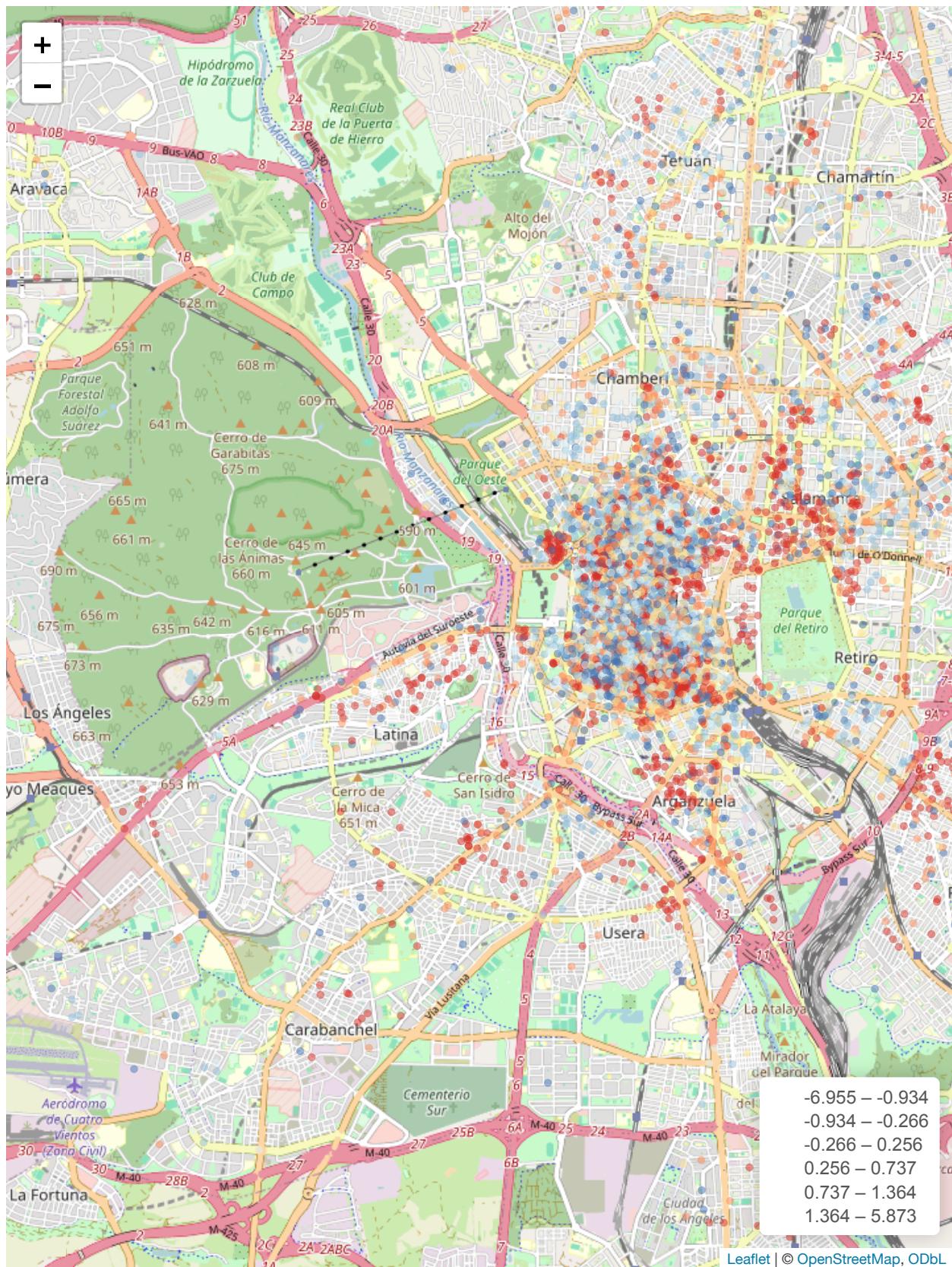
Gráfico I Moran



Podemos observar que obtenemos un valor del estadístico **I-Moran** de **6.719334e-02** lo que indica una leve correlación espacial positiva para los residuos del modelo lineal planteado, además podemos notar que por el **p-value** que obtenemos (**2.2e-16**), el estadístico es altamente significativo, por lo que podemos descartar la hipótesis nula, que sostiene que no existe correlación espacial entre los residuos del modelo.

Vemos si existe dependencia espacial local

```
local_moran <- as.data.frame(localmoran(x = df_nd$logprice, listw = nb2listw(nb, style="W")))
pl_pt(df_nd, color2 = local_moran$Z.Ii, size2=1, dd = 6)
```



Del análisis visual podemos ver apreciar ciertas zonas de sobrecarga de puntos rojos lo que nos podría estar

dando indicios de la existencia de dependencia espacial local, sobre todo en zona noroeste lindero al Parque del Rertiro.

Apartado 3

Para el desarrollo de este apartado obtendremos del paquete `osm` la ubicación de los bancos de la ciudad de Madrid, con ellos estableceremos cual es el mas cercano a cada domicilio e incorporaremos dicha métrica a nuestro dataset para luego analizar si persisten las correlaciones espaciales en nuestro modelo

```
# obtenemos los bancos de Madrid
mapa <- opq(bbox = "Madrid")
poligonos <- add_osm_feature(mapa, key = "amenity", value = "bank")
sf <- osmdata_sp(poligonos)

# convertimos los dataframes de casas y bancos a objetos sf
pisos <- st_as_sf(df_nd, coords = c("longitude", "latitude"), crs = 4326) # 4326 es CRS de WGS 84
bancos <- st_as_sf(sf$osm_points, coords = c("longitude", "latitude"), crs = 4326)

# pander(dim(bancos)) // _1634_ and _85_>

# calculamos las distancias en el producto cruzado
distancias <- st_distance(pisos, bancos)

# encontramos el banco mas cercano a cada casa
indice_banco_mas_cercano <- apply(distancias, 1, which.min)

# añadimos al dataframe las columnas banco mas cercano y la distancia al mismo
pisos <- pisos %>%
  mutate(distancia_minima = apply(distancias, 1, min)) # Añade distancia mínima

df_new <- st_drop_geometry(pisos)

df_new$longitude <- st_coordinates(pisos)[, 1] # Extraer longitud
df_new$latitude <- st_coordinates(pisos)[, 2] # Extraer latitud

dim(df_new)

## [1] 7619   26

modelo_2 <- lm(logprice~.-X -price, data=df_new)

# visualizamos resumen del modelo
pander(summary(modelo_2))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-53	42,33	-1,252	0,2105
minimum_nights	-0,0208	0,003844	-5,412	6,421e-08
number_of_reviews	0,0006168	0,0001151	5,36	8,564e-08
review_scores_value	0,03824	0,005402	7,078	1,597e-12
calculated_host_listings_count	-0,001296	0,0001511	-8,573	1,213e-17
bedrooms	0,1722	0,01066	16,15	9,871e-58
reviews_per_month	-0,1317	0,005766	-22,84	1,034e-111
beds	0,01563	0,005811	2,69	0,007151
accommodates	0,05254	0,004835	10,87	2,614e-27
availability_30	0,008152	0,001736	4,695	2,717e-06

	Estimate	Std. Error	t value	Pr(> t)
availability_60	-0,004709	0,001973	-2,386	0,01704
availability_90	0,0005495	0,000921	0,5967	0,5507
instant_bookable	0,0273	0,01131	2,415	0,01578
Distancia_Centro	-0,1041	0,006884	-15,12	6,695e-51
Distancia_Norte	0,02882	0,007797	3,696	0,0002209
Distancia_Sur	0,09187	0,01134	8,102	6,229e-16
tv_ports	0,00513	0,004468	1,148	0,251
phone_ports	0,00362	0,00447	0,8098	0,4181
Vecinos	0,008563	0,006134	1,396	0,1628
Piso	-0,00116	0,002932	-0,3956	0,6924
ventanas	-0,003959	0,004504	-0,879	0,3795
distancia_minima	-0,0001614	4,182e-05	-3,86	0,0001144
longitude	-1,464	0,4581	-3,197	0,001395
latitude	1,26	1,071	1,177	0,2393

Table 4: Fitting linear model: logprice ~ . - X - price

Observations	Residual Std. Error	R ²	Adjusted R ²
7.619	0,4357	0,3571	0,3551

Nuestro nuevo modelo nos entrega un R_a^2 de **0,3551** y un Error Std de los Residuos de **0,4357**.

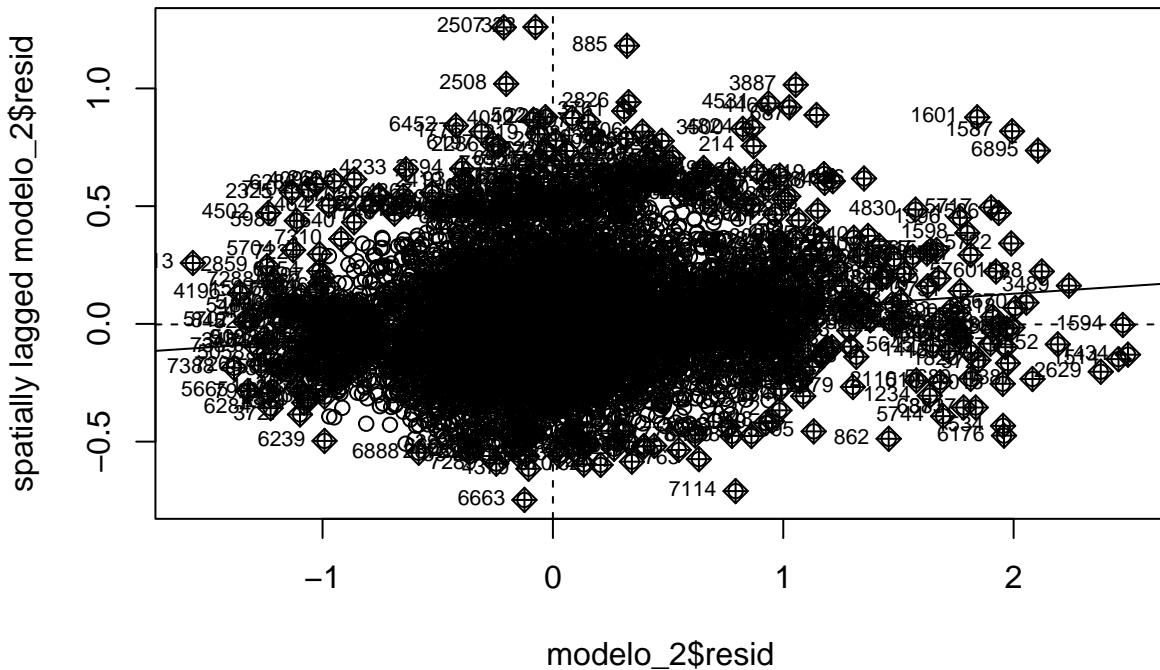
Ahora veamos si existe dependencia espacial global

```
# creamos la matriz de vecinos espaciales
nb_new <- knn2nb(knearneigh(cbind(df_new$longitude, df_new$latitude), k=5))

# obtenemos el test I-Moran
moran.test(x = modelo_2$resid, listw = nb2listw(nb_new, style="W"))

##
## Moran I test under randomisation
##
## data: modelo_2$resid
## weights: nb2listw(nb_new, style = "W")
##
## Moran I statistic standard deviate = 9.5686, p-value < 2.2e-16
## alternative hypothesis: greater
## sample estimates:
## Moran I statistic      Expectation      Variance
##       6.513400e-02     -1.312680e-04    4.652292e-05
moran.plot(x = modelo_2$resid, listw = nb2listw(nb_new, style="W"), main="Gráfico I Moran")
```

Gráfico I Moran



En este nuevo modelo habiendo incorporado las distancias mínimas a entidades bancarias aún presenta un **p-value** que nos indica que el estdístico I-Moran es significativo con un valor de **6.513400e-02** indicando dependencia espacial global.

Apartado 4

Como punto de partida dividiremos nuestro set de datos en dos grupos, entrenamiento (train) y testeo (test), en parte para luego poder hacer tests de validación cruzada y además para reducir lo costos computacionales en el calculo del modelo SAR

```
# dividimos nuestro set de datos para ahorrar tiempo de procesamiento
index <- sample(1:nrow(df_nd), 1000, replace = FALSE)
df_train <- df_nd[index,]
df_test <- df_nd[ -index, ]

# creamos la matriz de vecinos espaciales
nb_train <- knn2nb(knearneigh(cbind(df_train$longitude, df_train$latitude), k=5))

# definimos la formula simplificada de nuestros modelos
formula <- as.formula("logprice ~ .-X -price")

# definimos el modelo lineal
modelo_lm <- lm(formula, df_train)

# definimos el modelo espacial
modelo_espacial_sar <- lagsarlm(formula = formula, data = df_train, listw = nb2listw(nb_train, style="W"))

# visualizamos los summary de los modelos
summary(modelo_lm)
```

```

## 
## Call:
## lm(formula = formula, data = df_train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.34403 -0.25514 -0.02237  0.22999  2.16240
## 
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                -1.098e+02  1.030e+02 -1.066  0.28665  
## longitude                  7.624e-02  1.171e+00  0.065  0.94809  
## latitude                   2.807e+00  2.609e+00  1.076  0.28225  
## minimum_nights              -1.074e-03 1.056e-02 -0.102  0.91899  
## number_of_reviews            4.525e-04 3.102e-04  1.459  0.14492  
## review_scores_value         2.592e-02  1.545e-02  1.678  0.09363 .  
## calculated_host_listings_count -1.154e-03 3.977e-04 -2.901  0.00380 ** 
## bedrooms                    1.833e-01  2.847e-02  6.437 1.90e-10 *** 
## reviews_per_month            -1.036e-01 1.529e-02 -6.775 2.15e-11 *** 
## beds                        4.015e-02  1.699e-02  2.363  0.01831 *  
## accommodates                 4.117e-02 1.294e-02  3.182  0.00151 ** 
## availability_30              6.888e-03 4.850e-03  1.420  0.15583  
## availability_60              -1.626e-03 5.198e-03 -0.313  0.75441  
## availability_90              -1.973e-03 2.389e-03 -0.826  0.40923  
## instant_bookablet            5.179e-03 3.051e-02  0.170  0.86524  
## Distancia_Centro             -9.510e-02 1.875e-02 -5.073 4.68e-07 *** 
## Distancia_Norte               2.799e-02 1.887e-02  1.483  0.13829  
## Distancia_Sur                 6.046e-02 2.904e-02  2.082  0.03760 *  
## tv_ports                      8.561e-03 1.194e-02  0.717  0.47349  
## phone_ports                   3.158e-02 1.211e-02  2.608  0.00924 ** 
## Vecinos                       1.101e-02 1.645e-02  0.670  0.50331  
## Piso                          1.117e-02 8.028e-03  1.392  0.16433  
## ventanas                      4.444e-03 1.230e-02  0.361  0.71801  
## --- 
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4227 on 977 degrees of freedom
## Multiple R-squared:  0.4015, Adjusted R-squared:  0.388 
## F-statistic: 29.79 on 22 and 977 DF,  p-value: < 2.2e-16
summary(modelo_espacial_sar)

## 
## Call:lagsarlm(formula = formula, data = df_train, listw = nb2listw(nb_train,
##     style = "W"))
## 
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.283734 -0.242451 -0.025971  0.223675  2.153686
## 
## Type: lag
## Coefficients: (asymptotic standard errors)
##                               Estimate Std. Error z value Pr(>|z|)    
## (Intercept)                -7.3301e+01  1.0129e+02 -0.7237  0.469280  
## longitude                  1.2392e-01  1.1478e+00  0.1080  0.914023

```

```

## latitude           1.8928e+00  2.5653e+00  0.7378  0.460621
## minimum_nights    -5.4558e-04 1.0344e-02 -0.0527  0.957937
## number_of_reviews  4.6323e-04 3.0421e-04  1.5227  0.127831
## review_scores_value 2.3312e-02 1.5137e-02  1.5401  0.123538
## calculated_host_listings_count -1.1630e-03 3.8966e-04 -2.9847  0.002839
## bedrooms          1.7895e-01 2.7897e-02  6.4148  1.410e-10
## reviews_per_month -1.0266e-01 1.4984e-02 -6.8513  7.318e-12
## beds              4.0290e-02 1.6646e-02  2.4204  0.015502
## accommodates      4.1395e-02 1.2678e-02  3.2652  0.001094
## availability_30   7.0297e-03 4.7537e-03  1.4788  0.139198
## availability_60   -1.6544e-03 5.0927e-03 -0.3249  0.745294
## availability_90   -2.0356e-03 2.3410e-03 -0.8695  0.384556
## instant_bookablet -1.0124e-03 2.9901e-02 -0.0339  0.972990
## Distancia_Centro   -8.4393e-02 1.8618e-02 -4.5328  5.821e-06
## Distancia_Norte    2.1414e-02 1.8566e-02  1.1534  0.248749
## Distancia_Sur     5.4586e-02 2.8490e-02  1.9160  0.055367
## tv_ports           6.1170e-03 1.1697e-02  0.5230  0.600994
## phone_ports        3.1243e-02 1.1864e-02  2.6334  0.008453
## Vecinos            1.1405e-02 1.6116e-02  0.7077  0.479125
## Piso               1.0098e-02 7.8683e-03  1.2834  0.199368
## ventanas           3.7661e-03 1.2056e-02  0.3124  0.754752
##
## Rho: 0.15529, LR test value: 13.797, p-value: 0.00020369
## Asymptotic standard error: 0.041015
## z-value: 3.7863, p-value: 0.00015293
## Wald statistic: 14.336, p-value: 0.00015293
##
## Log likelihood: -539.2929 for lag model
## ML residual variance (sigma squared): 0.17151, (sigma: 0.41414)
## Number of observations: 1000
## Number of parameters estimated: 25
## AIC: 1128.6, (AIC for lm: 1140.4)
## LM test for residual autocorrelation
## test value: 2.9842, p-value: 0.084079
# obtenemos la suma de los residuos al cuadrado
sum((modelo_lm$resid)**2)

## [1] 174.5586
sum((modelo_espacial_sar$residuals)**2)

## [1] 171.513
# obtenemos los AIC de nuestros modelos
AIC(modelo_lm)

## [1] 1140.383
AIC(modelo_espacial_sar)

## [1] 1128.586

```

Podemos notar que no obtenemos una gran mejora en el modelo SAR respecto al modelo Lineal original. El **AIC** obtenido por el modelo SAR es levemente mejor que el del modelo Lineal, con un valor de **1200** vs los **1204** del modelo **glm**, lo que nos indica una mejor calidad para el modelo espacial. En tanto en la suma de los residuos al cuadrado tambien se percibe una mejora en el modelo espacial, con un valor de **184,5** vs los **186,1** del modelo lineal.

Apartado 5

```
modelo_espacial_sem <- errorsarlm(formula = formula, data = df_train, listw = nb2listw(nb_train, style=summary(modelo_espacial_sem))

##
## Call:errorsarlm(formula = formula, data = df_train, listw = nb2listw(nb_train,
##   style = "W"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.25620 -0.24487 -0.02446  0.21612  2.09255
##
## Type: error
## Coefficients: (asymptotic standard errors)
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -9.7081e+01 1.2012e+02 -0.8082 0.4189918
## longitude              3.5141e-01 1.3343e+00  0.2634 0.7922686
## latitude               2.5185e+00 3.0417e+00  0.8280 0.4076875
## minimum_nights         -3.3290e-05 1.0307e-02 -0.0032 0.9974230
## number_of_reviews       3.4039e-04 3.0162e-04  1.1285 0.2591057
## review_scores_value    2.1498e-02 1.5014e-02  1.4319 0.1521754
## calculated_host_listings_count -1.1685e-03 3.9359e-04 -2.9689 0.0029883
## bedrooms                1.7605e-01 2.7753e-02  6.3434 2.247e-10
## reviews_per_month        -9.9383e-02 1.4835e-02 -6.6992 2.096e-11
## beds                     4.1591e-02 1.6458e-02  2.5271 0.0115014
## accommodates             4.3162e-02 1.2575e-02  3.4323 0.0005985
## availability_30          6.0553e-03 4.7497e-03  1.2749 0.2023520
## availability_60          -1.3275e-03 5.0763e-03 -0.2615 0.7937062
## availability_90          -2.0545e-03 2.3296e-03 -0.8819 0.3778188
## instant_bookablet        -2.8243e-03 2.9918e-02 -0.0944 0.9247894
## Distancia_Centro          -9.6341e-02 2.2242e-02 -4.3315 1.481e-05
## Distancia_Norte            2.6976e-02 2.2129e-02  1.2190 0.2228360
## Distancia_Sur              6.0653e-02 3.3490e-02  1.8111 0.0701309
## tv_ports                  1.9928e-03 1.1605e-02  0.1717 0.8636629
## phone_ports                3.1464e-02 1.1792e-02  2.6681 0.0076276
## Vecinos                   1.1902e-02 1.6003e-02  0.7438 0.4570060
## Piso                      9.8330e-03 7.7731e-03  1.2650 0.2058658
## ventanas                  4.5073e-03 1.1999e-02  0.3756 0.7071872
##
## Lambda: 0.2048, LR test value: 17.589, p-value: 2.7418e-05
## Asymptotic standard error: 0.04758
## z-value: 4.3042, p-value: 1.6757e-05
## Wald statistic: 18.526, p-value: 1.6757e-05
##
## Log likelihood: -537.3968 for error model
## ML residual variance (sigma squared): 0.17036, (sigma: 0.41275)
## Number of observations: 1000
## Number of parameters estimated: 25
## AIC: 1124.8, (AIC for lm: 1140.4)

sum((modelo_espacial_sem$residuals)**2)

## [1] 170.3605
```

Podemos observar que con el nuevo modelo SEM propuesto mejoramos levemente la calidad de nuestra predicción, obteniendo un **AIC** de **1195,1** y una suma de residuos al cuadrado de **184,0**.