Memoria

El precio del alquiler en España y agentes relacionados

Autor

- *Apellidos:* González Fornell
- *Nombre:* José Manuel
- *Correo:* josemanuelgonzalezfornell@gmail.com

Índice

1. Introducción

El precio de la vivienda es un factor económico que afecta a toda la población. Las personas que son arrendatarias dedican un porcentaje de su sueldo a pagar la renta para poder vivir de forma independiente. En ciertas ciudades españolas, este porcentaje a pagar es muy elevado, pudiendo en ocasiones suponer en más del 50% de los ingresos del inquilino. De esta forma, la economía del arrendatario, y en general su vida íntegra, queda comprometida por este gasto elevado mensual. En consecuencia, a este hecho, muchas personas se ven arrastradas a vivir compartiendo piso gran parte de su vida, a privarse de tener otros bienes materiales como un coche o incluso a tener que abstenerse de formar una familia.

El entendimiento de los factores que condicionan el precio del alquiler en España y su evolución en el tiempo puede ser un buen punto de partida para investigar en que zonas es más rentable para el desarrollo de la vida del arrendatario.

2. Objetivo

El objetivo de esta investigación es realizar un análisis exploratorio de datos (EDA, de sus siglas en inglés) del precio del alquiler en España en diferentes años y municipios para ver su evolución en el tiempo y la variación de precio dependiendo del lugar. Además, se tomarán en cuenta factores como la población, el turismo y la cantidad de pisos turísticos de la zona para comprobar si estos agentes intervienen de alguna manera en el precio del alquiler.

3. Hipótesis

- *Hipótesis primaria:*

Cuanta mayor es la población de una zona, más altos son los precios del alquiler por metro cuadrado.

- *Hipótesis secundarias:*
  - A mayor turismo internacional recibido, mayor es el precio de los alquileres por metro cuadrado.
  - El precio del alquiler por metro cuadrado aumenta a medida que pasa el tiempo.
  - La cantidad de pisos turísticos es directamente proporcional a la cantidad de turismo internacional recibido en una zona.
  - En el caso de que la cantidad de pisos turísticos aumente, el precio del alquiler por metro cuadrado también aumentará.
  - El municipio con el precio del alquiler más alto es el municipio con mayor población.
  - Hay una diferencia significativa entre el precio del alquiler por metro cuadrado de la vivienda colectiva y el precio del alquiler de la vivienda unifamiliar o rural.
  - Existe más pisos turísticos que pisos de alquiler y esta diferencia es dependiente del turismo.

4. Exploración inicial y preparación de datos

4.1. Importación de librerías

Lo primero que se realiza es la importación de las librerías necesarias para realizar el análisis. En este caso las librerías utilizadas son las siguientes:

- Pandas. Importada con el pseudónimo "pd".
- Numpy. Importada con el pseudónimo "np".
- Regular expresion.
- Seaborn. Importada con el pseudónimo "sns".
- Functions. Librería propia importada con pseudónimo "fn"
- Matplotlib.pyplot. Importada con el pseudónimo "plt".
- Scipy.stats. Importada con el pseudónimo "ss".
- La función filterwarnings() de la librería warnings.

4.2. Importación de bases de datos

Se importan las bases de datos con las que se van a trabajar.

Los datasets usados han sido obtenidos a través de la página del Instituto Nacional de Estadística (INE). Se han usado varios datasets para poder corroborar las hipótesis propuestas anteriormente:

1. Datos de viviendas en alquiler por municipios:

   Cantidad de viviendas en alquiler por municipios en España desde 2015 hasta 2021, junto con sus precios expuestos de diferentes formas. Además, este Dataset clasifica las viviendas en dos conjuntos dependiendo de la superficie de esta: Colectiva y Unifamiliar o Rural.

   Los datos han sido obtenidos a partir de las tributaciones de los contribuyentes, por lo que no incluye los datos de los municipios pertenecientes a los fueros de País Vasco ni de la Comunidad Foral de Navarra.

   Link del Dataset

2. Relación de municipios con Comunidades autónomas:

   Relaciona los municipios de España con las comunidades autónomas a las que pertenecen.

   Link del Dataset

3. Población por municipios españoles.

   Población de los diferentes municipios españoles en total y por tramos de edad desde 2003 hasta 2022. También se incluye en este dataset los datos respecto a la población total nacional.

   Link del Dataset

4. Censo de viviendas en España.

   Cantidad de edificios dedicados principal o exclusivamente a la vivienda en municipios de más de 2.000 habitantes hasta 2011 (fecha del último censo de viviendas realizado). En este Dataset se muestra además el estado de la vivienda y el año de construcción.

   Link del Dataset

5. Censo de viviendas turísticas en España.

   Cantidad de alquileres turísticos por municipios producidos en España en los meses de febrero y de agosto en los años 2020, 2021 y 2022. También se muestra las viviendas, las plazas por viviendas y solo las plazas turísticas.

Este Datasets fue obtenido mediante una técnica de Web Scraping en las 3 mayores páginas de alquiler turístico utilizadas.

Link del Dataset

6. Turismo internacional en España.

   Cantidad de turismo internacional recibida en España en los años 2020, 2021 y 2022 por municipio.

   Estos datos se obtuvieron a partir de la geolocalización de dispositivos de telefonía móvil.

   Link del Dataset

4.3. Primera exploración de los DataFrame

Tras la importación, se realiza una primera exploración de cada dataframe. De esta manera se podrá entender el funcionamiento y estructura de estos y se podrá saber cómo trabajar con ellos.

Inicialmente se observa el número de filas y el número de columnas de los dataframes usando el atributo **shape**. También se visualizan los índices de columna del dataframe y los tipos de datos y la cantidad de valores no nulos contenidos en cada columna usando el método **info()**. En caso de que el dataframe contenga tantas columnas que no nos indique los valores no nulos, se usa el método **count()** para este mismo fin.

Seguidamente, se observa las 5 primeras filas usando el método **head()**. Esto permite ver el formato de escritura de los datos y hacerse una idea de cómo está estructurado el dataframe.

Finalmente, se realiza un pequeño e inicial análisis estadístico de los datos usando el método **describe()**.

4.4. Depuración de bases de datos

Tras la primera exploración y toma de contacto con las bases de datos a trabajar, se procede a realizar la depuración de los datos. En este proceso se eliminarán los datos no deseados y se tratarán los valores nulos de las bases de datos.

4.4.1 Base de datos de alquileres por municipio

Se revisa la documentación de esta base de datos puesto que los títulos de las columnas no son autoexplicativos (no es el caso de las demás base de datos con las que se trabajarán, por lo que esto no se hará con el resto). A continuación, en la *Tabla 1* se detalla el significado del título de cada columna y el tipo de valor que contiene cada columna:

Tabla 1. *Leyenda de los nombre de cada columna de la base de datos de alquileres por municipio y los tipos de valores que contiene cada columna. La denominación del campo se marca al final con "_AA" para señalar el año*

Código	**Significado**	**Tipo de valor**
CPRO	Código de Provincia Censo Población 2011 INE (numérico)	INT
NPRO	Nombre de Provincia Censo Población 2011 INE	VARCHAR(50)
CUMUN	Código único de Municipio Censo Población 2011 INE (numérico)	INT
NMUN	Nombre de Municipio Censo Población 2011 INE	VARCHAR(50)
BI_ALVHEPCO_TVC_AA	Recuento subconjunto alquiler Grupo GGT01 VC: Vivienda Colectiva como tipología de más superficie	INT
BI_ALVHEPCO_TVU_AA	Recuento subconjunto alquiler Grupo GGT01 VU: Vivienda Unifamiliar o Rural como tipología de más superficie	INT
ALQM2mes_LV_M_VC_AA	Alquiler mensual m2 mediano m2 Grupo GGT01 VC: Vivienda Colectiva como tipología de más superficie	FLOAT
ALQM2mes_LV_25_VC_AA	Alquiler mensual m2 percentil 25 Grupo GGT01 VC: Vivienda Colectiva como tipología de más superficie	FLOAT
ALQM2mes_LV_75_VC_AA	Alquiler mensual m2 percentil 75 Grupo GGT01 VC: Vivienda Colectiva como tipología de más superficie	FLOAT
ALQM2mes_LV_M_VU_AA	Alquiler mensual m2 mediano m2 Grupo GGT01 VU: Vivienda Unifamiliar o Rural como tipología de más superficie	FLOAT
ALQM2mes_LV_25_VU_AA	Alquiler mensual m2 percentil 25 Grupo GGT01 VC: Vivienda Unifamiliar o Rural como tipología de más superficie	FLOAT

ALQM2mes_LV_75_VU_AA	Alquiler mensual m2 percentil 75 Grupo GGT01 VC: Vivienda Unifamiliar o Rural como tipología de más superficie	FLOAT
ALQTBID12_M_VC_AA	Alquiler mensual todo el bien inmueble mediana Grupo GGT01 VC: Vivienda Colectiva como tipología de más superficie	FLOAT
ALQTBID12_25_VC_AA	Alquiler mensual todo el bien inmueble percentil 25 Grupo GGT01 VC: Vivienda Colectiva como tipología de más superficie	FLOAT
ALQTBID12_75_VC_AA	Alquiler mensual todo el bien inmueble percentil 75 Grupo GGT01 VC: Vivienda Colectiva como tipología de más superficie	FLOAT
ALQTBID12_M_VU_AA	Alquiler mensual todo el bien inmueble mediana Grupo GGT01 VU: Vivienda Unifamiliar o Rural como tipología de más superficie	FLOAT
ALQTBID12_25_VU_AA	Alquiler mensual todo el bien inmueble percentil 25 Grupo GGT01 VC: Vivienda Unifamiliar o Rural como tipología de más superficie	FLOAT
ALQTBID12_75_VU_AA	Alquiler mensual todo el bien inmueble percentil 75 Grupo GGT01 VC: Vivienda Unifamiliar o Rural como tipología de más superficie	FLOAT
SLVM2_M_VC_AA	Superficie m2 mediana Grupo GGT01 VC: Vivienda Colectiva como tipología de más superficie	FLOAT
SLVM2_25_VC_AA	Superficie m2 percentil 25 Grupo GGT01 VC: Vivienda Colectiva como tipología de más superficie	FLOAT
SLVM2_75_VC_AA	Superficie m2 percentil 75 Grupo GGT01 VC: Vivienda Colectiva como tipología de más superficie	FLOAT

SLVM2_M_VU_AA	Superficie m2 mediana Grupo GGT01 VU: Vivienda Unifamiliar o Rural como tipología de más superficie	FLOAT
SLVM2_25_VU_AA	Superficie m2 percentil 25 Grupo GGT01 VU: Vivienda Unifamiliar o Rural como tipología de más superficie	FLOAT
SLVM2_75_VU_AA	Superficie m2 percentil 75 Grupo GGT01 VU: Vivienda Unifamiliar o Rural como tipología de más superficie	FLOAT

De esta base de datos, solo se requieren las columnas que contienen los siguientes datos:

- Códigos de provincia (**CPRO**)
- Nombre de la provincia (**NPRO**)
- Código único de municipio (**CUMUN**)
- Nombre de municipio (**NMUN**)
- Recuento de alquileres tanto de vivienda colectiva como de vivienda unifamiliar o rural (**BI_ALVHEPCO_TVC_AA** y **BI_ALVHEPCO_TVU_AA**)
- Mediana del alquiler mensual por metro cuadrado en viviendas colectivas y unifamiliares o rurales (**ALQM2mes_LV_M_VC_AA** y **ALQM2mes_LV_M_VU_AA**)

El resto de columnas que no contienen datos de interés para el propósito de este EDA son eliminadas. Para ello se realiza un bucle **for** en las columnas del dataframe con el atributo **columns** para recorrer los índices de las columnas. Dentro del bucle, usando un condicional **if** y expresiones regulares se eliminan las columnas que no coinciden con nuestras expresiones regulares o nuestros títulos deseados usando el método **drop**.

Seguidamente, se determina la columna **NMUN** como índice del DataFrame, ya que, a partir de este momento, se trabajará con los nombres de municipios como índice de todas las bases de datos. Para ello se utiliza el método **set_index()**.

Posteriormente, se eliminan las columnas y las filas donde todos los valores son nulos, puesto que estas filas no son de utilidad. A la hora de eliminar las filas no se tendrán en cuenta las columnas **CPRO_A**, **NPRO** ni **CUMUN_A** debido a que una instancia, aunque tenga valores no nulos en estas columnas, si en el resto de columnas todos los valores son nulos, estas instancias no contienen datos de relevancia. Para esta tarea se utiliza el método **dropna()**.

Después, se cambia la forma en la que se indican los años en los índices de las columnas para que aparezcan 4 dígitos en vez de dos. Para ello se usa un bucle **for**

sobre el índice de columnas del DataFrame usando el atributo **columns**. Dentro del bucle, se usa el método **replace()** junto con expresiones regulares para obtener el cambio de formato de los años.

Finalmente, los valores nulos son cambiados por la media de la provincia usando el método **fillna()**, junto con la función **groupby()** y **transform()** para obtener la media por provincias. Son cambiados por estos datos debido a que, de esta manera, no se obtienen outliers y el análisis se ajusta más a la realidad.

4.4.2 Base de datos de población por municipios

De esta base de datos únicamente son de interés para el objetivo de este EDA los valores de población totales, sin estar clasificado por sexos ni por edades, de todos los municipios. Por ello se realiza una máscara para filtrar estos datos y desechar el resto.

Una vez obtenidos los datos deseados, se procede a cambiar el separador decimal de la columna **TOTAL** de coma a punto, con el método **replace()**, y a cambiar el formato de los datos de esta columna a float, con el método **astype()**. También se cambian los datos de la columna **Periodo** paraque muestre únicamente el año del periodo y se cambia el formato de los datos de esta columna a datetime64[Y] usando el método **astype()**. Además, para que únicamente se muestre el año, se usa el atributo **dt.year**. Además, se elimina de la columna **Municipios** todo lo que no sea el nombre del municipio usando el método **replace()** con expresiones regulares.

Finalmente, se realiza una tabla pivote, usando el método **pivot_table()**, para obtener una tabla que tenga como índice los municipios y como columnas las poblaciones totales de cada municipio desglosada por años. Además, se utiliza el método **add_prefix()** para añadir el prefijo **Población_** a los índices de columnas.

4.4.3 Base de datos de viviendas

De esta base de datos únicamente son de interés los datos de los inmuebles totales por municipios. Por ello, se usa un filtro para obtener únicamente estos datos. Además, se requiere que se muestren los nombres de los municipios como índice y los inmuebles totales como columna. Para obtener un DataFrame con esas características, antes del filtrado se determina como índice del DataFrame la columna **Municipios (con más de 2.000 habitantes)** usando el método **set_index()** y junto al filtrado se determina que solo se muestre la columna **Total**.

También se cambia el nombre de la columna **Total** a **Inmuebles_totales** usando el método **rename()**.

Una vez obtenido un DataSet con los datos deseados, cambiamos el índice para que únicamente aparezca el nombre de los municipios, sin el código postal, usando la función **replace()** junto con expresiones regulares. Esto se realiza para que el índice tenga el mismo formato que los demás DataSet que han sido depurados y el futuro merge sea más sencillo.

Además, usando la misma función que anteriormente, se elimina el separador millares de la columna **Inmuebles_totales**. También se cambia el formato de los datos a float usando el método **astype()**.

4.4.4 Base de datos de viviendas turísticas

En esta base de datos, únicamente son de interés las viviendas turísticas por municipio clasificadas por año. Por ello se realiza un filtro para obtener dichos datos.

Tras obtener los datos deseados, se les aporta el formato deseado. Inicialmente, al igual que se ha realizado anteriormente en el resto de DataSets, se elimina de la columna **Municipios** todo aquello que no sea el nombre del municipio usando la función **replace()** junto con expresiones regulares. Posteriormente, se elimina el separador de millares de la columna **Total** usando el mismo método. Además, se formatea los datos de esta columna a enteros usando el método **astype()**. También se modifica la columna **Periodo** para que tenga un formato datetime64[Y] usando los métodos anteriormente menciones. Para que la columna periodo muestre únicamente el año, se usa el atributo **dt.year**.

Finalmente, se realiza una tabla pivote, usando el método **pivot_table()**, que contenga los municipios como índice y los valores totales de viviendas turísticas divididos por año. Al producir esta tabla pivote se usa la función de agregado suma para sumar las viviendas turísticas de todos los meses de cada año. Además, se le agrega el prefijo **Viviendas_turisticas_** a los índices de columna.

4.4.5 Base de datos de turismo

Los datos del turismo están divididos en 3 DataFrames diferentes para cada uno de los años. Sin embargo, el formato de estos 3 DataFrame es el mismo, por lo que para trabajar con ellos se creará una lista con los tres DataSets y se recorrerá dicha lista con un bucle **for**, aplicando la modificaciones necesarias.

Lo primero que se realiza es el formateo de la columna **mes**, aplicándole un formato datetime64[Y] usando el método **astype()** y mostrando únicamente el año con el atributo **dt.year**. Seguidamente, se crea una tabla pivote, donde el índice es el municipio de destino y las columnas son los turistas totales por año. Esto se realiza usando la función **pivottable()** junto con la función de agregación **sum**. Tras la creación de la tabla pivote, se agrega el prefijo **Turistas_** a los índices de columna. Por último, se mergea todas las tablas pivotes creadas con el bucle usando el método **merge()**.

4.5. Merge y agregaciones

Una vez se depuran todos los datos de todas las bases de datos se procede a hacer un merge de todas las bases de datos. Primero, se realiza un merge de los datos de alquiler con la relación de CCAA con los municipios. Para este merge, toma de la base de datos de relación de CCAA con provincias únicamente las columnas **CPRO** y **Comunidad Autónoma**. Se realiza el merge, usando la función **merge()** fijándonos en la columna **CPRO** de la base de datos de relación de CCAA con provincias y en la columna **CPRO_A** de la base de datos del alquiler. Posteriormente, se elimina la columna **CPRO** del DataFrame resultante con el método **drop()** y se establece la columna **NMUN** como índice con la función **set_index()**.

El resto de merge se realizan utilizando también el método **merge()**, pero fijando únicamente los índices.

Tras obtener un DataSet con todos los datos deseados de las diferentes bases de datos, se procede a realizar las siguientes operaciones:

- Añadir columnas con el porcentaje de viviendas turísticas por año.
- Añadir columnas con el total de casas en alquiler por año.
- Añadir columnas con el porcentaje de viviendas en alquiler por año.
- Añadir columnas con el porcentaje de alquiler de viviendas colectivas por año.
- Añadir columnas con el porcentaje de alquiler de viviendas individuales y familiares por año.
- Renombrar el índice de columnas, usando el método **rename()**, para que tenga una leyenda autoexplicativa, sin espacios ni caracteres extraños.

Una vez obtenido el DataSet final deseado, se procede a crear otro DataSet con los mismos datos que el ya obtenido, pero únicamente con aquellos correspondientes a los años 2020 y 2021 usando la función **filter()** y la función **drop()**. Esto se realiza debido a que son los dos años donde coincide que existen todos los datos de todas las bases de datos.

4.6. Análisis exploratorio

Una vez obtenidos los dos DataSets con los que se trabajaran, se realiza un breve análisis inicial, como realizamos anteriormente con las otras bases de datos, para comprobar que hemos obtenido los DataSets correctamente.

Para realizar esta comprobación se usarán los métodos **info()**, **head()** y **describe()** y el atributo **shape**.

4.7. Tratamiento de outliers

Tras comprobar que los DataFrame obtenidos están correctamente se comprueba los outliers que pudieran existir, para ello se usa una función de creación propia llamada **get_outliers()**. Esta función recibe como parámetro un DataFrame y devuelve una lista con 2 diccionarios: El primero indica la cantidad de outliers en cada columna y el segundo los outliers de cada columna.

Tras observar que existen outliers, se crea un nuevo DataFrame donde los outliers que estén por encima del máximo se cambiarán a este máximo y los que estén por debajo del mínimo se cambiaran al mínimo. En este caso se usará otra función de creación propio denominada **change_outliers()**. En esta función, el atributo es un DataFrame y devuelve una copia con los outliers del DataFrame pasado como atributos cambiados.

Se comprueba que los outliers han sido eliminados usando nuevamente sobre el nuevo DataFrame la función **get_outliers()**.

Con el DataFrame sin outliers preparado, se repite el mismo proceso para los outliers de los años 2020 y 2021.

4.8. Exportación de los DataFrames

Finalmente, tras todo el proceso de depuración, se exportan los DataFrames obtenidos y tratados a un archivo csv, para poder utilizarlos posteriormente en otras bases de datos. Para ello se usa el método **to_csv()**, utilizando como separador ";".

5. Análisis estadístico

Una vez depurados y preparados los DataFrames con los que se trabajará, se procede a realizar el análisis estadístico y a corroborar las diferentes hipótesis propuestas.

5.1. Análisis univariante

Se realiza un análisis de los datos de cada columna. En este análisis se intentará conocer las diferentes variables contenidas en el dataframe. Para ello, se obtendrá la media, la mediana, la moda, la varianza, la desviación estándar, el percentil 25 y el percentil 75 . Además, se obtendrá un histograma de los datos para saber si los datos siguen una distribución normal y se corroborará este hecho con el test de hipótesis Kolmogorov-Smirnov. Por último, se realiza un boxplot para observar los outliers y el rango de los datos. Estos datos se obtienen usando los métodos **mean()**, **meadian()**, **moda()**, **var()**, **std()**, **quantile()**, **displot()**, **kstest()** y **boxplot()**. Por otro lado, las variables categóricas se analizarán mostrando los valores únicos y la cantidad de veces que aparecen y el número total de valores únicos que tiene dicha variable. Esto se realizará con los métodos **value_counts()** y **nunique()**. Todos estos métodos mencionados anteriormente están contenidos dentro de la función propia **get_univariate_analysis()**. Para las gráficas usamos el DataFrame sin outliers para mejorar la visualización.

Se observa que la mayoría de los variables contienen una gran cantidad de outliers. Además, también se puede determinar tan solo 8 variables parecen seguir una distribución normal, el resto no.

5.2. Análisis bivariante

Tras el análisis univariante, se debe realizar un análisis bivariante de las columnas principales del DataFrame. Las columnas que seleccionaremos son aquellas que corresponden a los datos de 2021 y la columna de **Inmuebles_totales**. Posteriormente, se realiza un heatmap y gráficos de dispersión por pares, de esta forma se obtendrá una idea de la correlación entre columnas. Para esta labor, se utilizan las funciones **heatmap()** y **pairplot()** de seaborn contenidas en una función propia denominada **get_bivariate_analysis()**.

Se puede observar en el análisis bivariante que existe una alta correlación entre los datos de viviendas colectivas con los datos de población e inmuebles totales, entre otros.

5.3 Análisis multivariante

Por último, se realiza un análisis multivariante de los datos. Para este análisis se realizará un gráfico de tarta donde se representarán la media de la cantidad de las viviendas turísticas, de las viviendas colectivas en alquiler y de las viviendas unifamiliares o rurales

en alquiler en 2021. Se realiza con estas columnas puesto que son las más interesantes. Con este análisis se visualiza cual de todos los tipos de viviendas es el que se encuentra en mayor cantidad. Para este análisis se utiliza el método **pie()** y las columnas **Total_vc_2021**, **Total_vu_2021** y **Viviendas_turisticas_2021**.

Se puede apreciar que las viviendas colectivas en alquiler son las mayoritarias, seguidas de las viviendas turísticas y, por último, en menor cantidad se encuentran las viviendas unifamiliares o rurales.

No se realizan más análisis multivariantes porque en el apartado siguiente, al responder a las hipótesis, se realizarán más.

5.4. Respondiendo las hipótesis

Una vez obtenido los análisis univariantes, bivariantes y multivariantes se comienzan a responder las hipótesis planteadas inicialmente. Las hipótesis se responderán una por una siguiendo el orden en el cual se presentaron al inicio del notebook.

5.4.1. Correlación entre alquiler y población

La primera hipótesis determina que existe una correlación entre el precio del alquiler y la población de cada municipio. Para comprobarlo, inicialmente se realiza un gráfico lineal de las columnas de **Alquiler_mes_vc_m2_2021** y **Poblacion_2021** del DataFrame final sin outliers para mejorar la visualización. Usamos los datos de 2021 porque son los últimos que tenemos. Para realizar esta gráfica se usa la función **lmplot()** de seaborn.

A simple vista, observando el gráfico lineal resultante, no parece existir una correlación, pero para asegurar este hecho, se realiza una correlación de spearman, usando el método **spearmanr()**. Se usa este método debido a que ambas variables no siguen una distribución normal. Tras usar este método, se obtiene el dato estadístico y el cual sirve para determinar si tienen correlación o no. Si este dato es superior a 0,5 se determinará que ambas variables tienen una fuerte correlación, en caso contrario se determinará que no existe correlación entre ambos datos. Se usa el dato estadístico y no el p valor porque con el p valor estaba dando problemas y datos que no tienen ninguna correlación aparecen como si las tuviera, por eso tomamos como referencia el valor estadístico en esta prueba de correlación y en el resto de pruebas de correlación del EDA.

Para este test se usan los datos con los outliers.

Este proceso se repite, pero para comparar las viviendas unifamiliares o rurales en vez de las viviendas colectivas. En este caso se usarán los datos presentes en las columnas **Alquiler_mes_vu_m2_2021** y **Poblacion_2021**

5.4.2. Correlación entre alquiler y turismo

Para corroborar esta hipótesis se realizan los mismos análisis que en el apartado anterior, solamente que usando las columnas **Alquiler_mes_vu_m2_2021** y **Turistas_2021**.

Este proceso se repite, pero para comparar las viviendas unifamiliares o rurales en vez de las viviendas colectivas. En este caso se usarán los datos presentes en las columnas **Alquiler_mes_vu_m2_2021** y **Turistas_2021**.

5.4.3. Aumento del alquiler con el tiempo

Para esta hipótesis, se toman las columnas que indican el precio del alquiler por metro cuadrado de cada año y se realiza un gráfico de barras de la media de estos precios, usando los métodos **bar()** y **mean()**. Además, se realiza un gráfico de línea usando la función **plot()** para observar la tendencia de la media en el tiempo. Para que las dos gráficas se vean superpuestas se usa la función **subplots()**.

Esta metodología se realiza primero con los precios de la vivienda colectiva, contenidos en las columnas que comienzan con el término **Alquiler_mes_vc**, y posteriormente con los precios de las viviendas unifamiliares o rurales, contenidos en las columnas que comienzan con el término **Alquiler_mes_vu**.

Visualmente, parece que el precio de la vivienda aumenta con el tiempo. Sin embargo, para comprobar que no todas las variables son iguales y que al menos una variable es significativamente diferente del resto, se realiza una prueba de Friedman, la cual sirve para comparar dos o más variables no paramétricas dependientes. Para esta labor se usa la función **friedmanchisquare()** de la librería **scipy.stats**, que está contenida dentro de la función propia denominada **get_significance_friedman()**.

Realizamos el mismo proceso que anteriormente, pero con las columnas que comienzan con el término **Alquiler_mes_vu**.

Comprobamos si hay diferencias significativas entre al menos uno de los grupos.

5.4.4. Correlación pisos turísticos y turismo

Esta hipótesis se puede corroborar usando la metodología utilizada en el punto **5.4.1**. Sin embargo, en este caso se usarán las columnas **Turistas_2021** y **Viviendas_turisticas_2021**.

5.4.5. Correlación pisos turísticos y alquiler

Al igual que en el apartado anterior, para corroborar esta hipótesis se debe utiliza la metodología utiliza en el punto **5.4.1**, pero en este caso las variables con las que se trabajarán pertenecerán a las columnas **Viviendas_turisticas_2021**, **Alquiler_mes_vc_m2_2021** y **Alquiler_mes_vu_m2_2021**. Primero se realizará el análisis con las viviendas colectivas y posteriormente con las viviendas unifamiliares o rurales.

5.4.6. El precio más alto es del municipio de mayor población

Para esta hipótesis, se toma el municipio con mayor población y el municipio con mayor alquiler de la vivienda y se compara. Para ello se utiliza la función **nlargest()**.

Primero se compara el precio de la vivienda colectiva, columna **Alquiler_mes_vc_m2_2021**, con la población, columna **Poblacion_2021**. Posteriormente, se compara con la población el precio de la vivienda unifamiliar o rural, columna **Alquiler_mes_vu_m2_2021**.

Repetimos el proceso para las viviendas unifamiliares o rurales.

5.4.7. Diferencia significativa entre alquiler de vivienda colectiva y alquiler de vivienda unifamiliar o rural

Para comparar estas dos variables, inicialmente se procede a realizar una prueba más visual y se realizan dos diagramas de cajas usando la función **boxplot()** de la librería de **seaborn**. En este caso, se aplica la función a las columnas **Alquiler_mes_vc_m2_2021** y **Alquiler_mes_vu_m2_2021**.

A simple vista, se observa que hay diferencia de precio entre ambos tipos de viviendas, no obstante, no existe mucha, por lo que se realizará un test de Mann-Whitney para confirmar si esta hipótesis es cierta o no. Se realiza este test porque ambas variables son independientes y no paramétricas y se utiliza la función **mannwhitneyu()** de la librería **scipy.stats**.

5.4.8. Diferencia entre pisos turísticos y pisos de alquiler dependiente del turismo

Para comprobar esta hipótesis, inicialmente se realiza un boxplot de las columnas **Viviendas_turisticas_2021**, **Total_casas_alquiler_2021** y **Turistas_2021**, usando la función **boxplot()**, para comprobar visualmente si existe una diferencia significativa.

Además, se realiza un pair plot de estas columnas usando la función **pairplot()** y se comprueba si existe una correlación entre ellas usando la función **spearmanr()** para realizar un test de Spearman del mismo modo que lo hemos realizado anteriormente.

Aparentemente, se observa una diferencia significativa entre ambos grupos y una fuerte correlación entre ellos, sin embargo, se comprueba esta hipótesis y si esta posible diferencia depende de la columna **Turistas_2021** analizando si existe una diferencia significativa entre ambas columnas mediante un test de Mann-Whitney usando la formula **mannwhitneyu()**. En caso de que no exista una diferencia significativa se explora si existe una diferencia significativa entre estas columnas con la variable **Turistas_2021** usando un test de Kruskal-Wallis mediante la función **kruskal()**.

Tras esta comprobación, se obtienen los valores únicos de la variable **Turistas_2021** usando la función **unique()**, y los valores de **Viviendas_turisticas_2021** y **Total_casas_alquiler_2021** para cada valor único de la columna de turistas. Se hace un test de Mann-Whitney para los datos de estas dos últimas variables con cada valor único de la variable de turistas, usando de nuevo la fórmula **mannwhitneyu()**.

Finalmente, se observan los p valores de este último análisis obtenido y se comprueba si existen diferencias significativas entre ambas columnas dependiente de la columna **Turistas_2021**.

Todos estos análisis están contenidos dentro de la función propia **get_difference_in_terms()**. En esta función, los gráficos se realizarán con los datos sin outliers para mejorar la visualización, sin embargo, el resto de análisis se realizan con los datos con outliers.

6. Conclusiones

Las conclusiones obtenidas con el análisis exploratorio de datos son:

- **No** existe una correlación entre el **precio del alquiler por metro cuadrado** y la **población** de los municipios.
- **No** existe una correlación entre el **precio del alquiler por metro cuadrado** y la cantidad de **turismo** recibido.
- El **precio del alquiler por metro cuadrado aumenta** con el tiempo de forma significativa.
- **Existe** una correlación entre la cantidad de **turismo** recibido y la cantidad de **viviendas turísticas**.
- **No** existe una correlación entre el **precio del alquiler por metro cuadrado** y la cantidad de **viviendas turísticas**.
- El **precio más alto** del alquiler por metro cuadrado **no** se da en el municipio con **mayor población**.
- El **precio del alquiler por metro cuadrado** de las viviendas **colectivas** es significativamente mayor que el de las viviendas **unifamiliares o rurales**.
- Existe una correlación entre las **viviendas turísticas** y las **casas totales en alquiler** y entre el **turismo** recibido y las **casas totales en alquiler**.
- La diferencia significativa que se da entre la cantidad de **viviendas turísticas** y la cantidad de **casas totales en alquiler no** depende del **turismo**. Sin embargo, la cantidad de **turismo** recibido **sí** depende de la cantidad de **viviendas turísticas** y la cantidad de **casas totales en alquiler**. Por tanto, esto quiere decir que la cantidad de **turismo** recibido depende de la cantidad de **viviendas turísticas** y la cantidad de **casas totales en alquiler,** pero no al revés.
