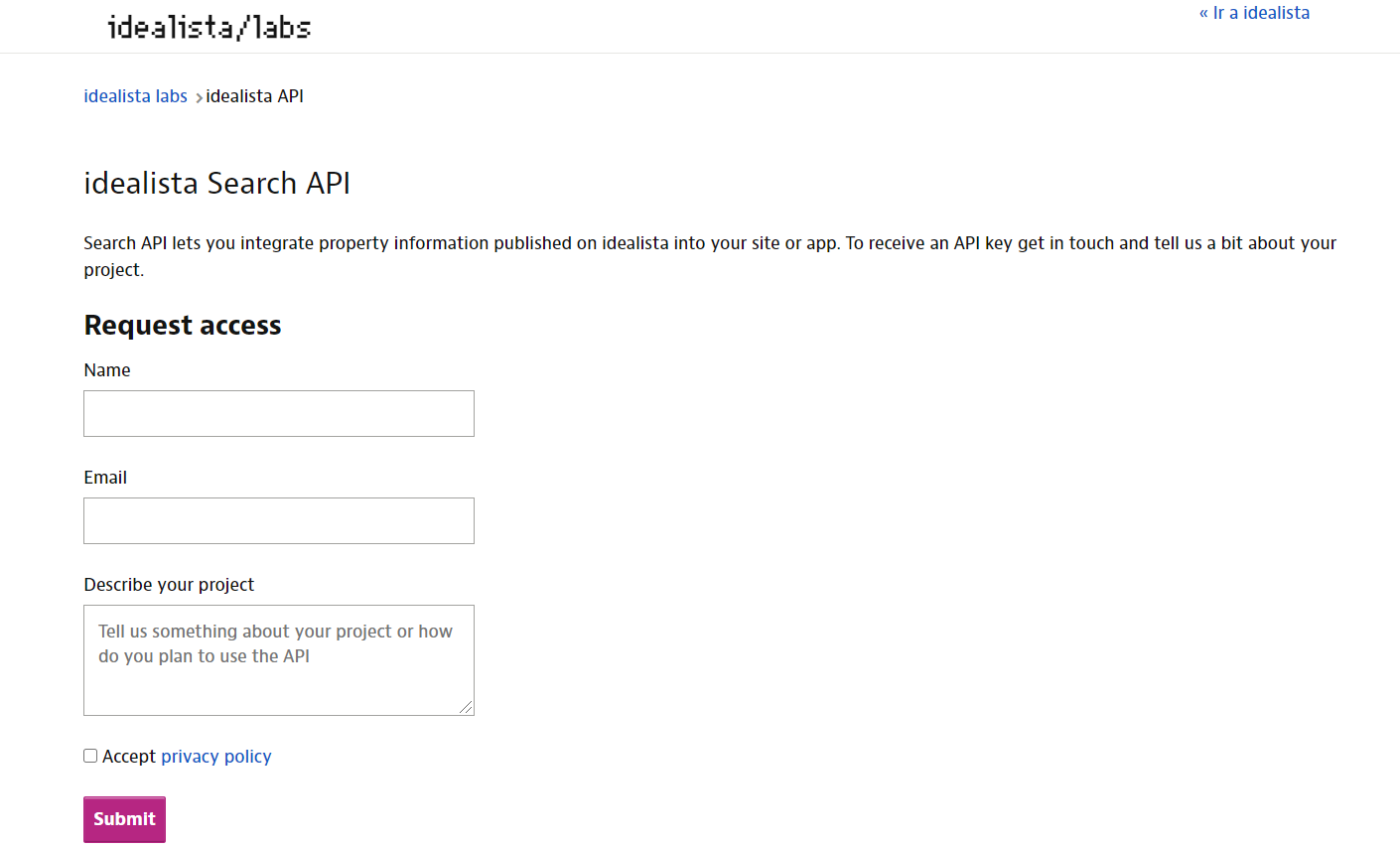
PREDICCION DEL PRECIO DE LA VIVIENDA EN VALLADOLID y BILBAO

El objetivo de este proyecto es predecir el precio de las viviendas en venta de la página **pisos.com** para las capitales de **Valladolid y** **Bilbao** en función de las siguientes características:

* Planta
* Metros
* Número habitaciones
* Número baños
* Tipo de vivienda (Casa, Chalet, Loft, Ático…)
* Latitud y Longitud 🡪 muchos valores nulos, decido generarlos
* **Distancia al centro 🡪 librería Geopy**
* **Precio en abril de 2020** 🡪 Interesante, uso en Tableau
* Ascensor (sí/no) 🡪 uso de Natural Language Processing?
* Garaje (sí/no) 🡪 uso de Natural Language Processing?
* Jardín (sí/no) 🡪 uso de Natural Language Processing?
* Obra nueva (sí/no)
* Barrio 🡪 uso de Natural Language Processing?
* Renta por persona por barrio

1. **OBTENCION DE DATOS**

Este es un paso crítico y con una influencia total sobre cómo será de adecuado el modelo que utilice para la predicción. Cuanto más y mejores datos obtenga, mejor será el rendimiento del modelo.

**Plan A:** API Idealista 

Inconveniente1: tras solicitar acceso a los tokens desde hace 5 días, no he recibido respuesta por mail

Inconveniente2: quiero ver la evolución del precio de abril de 2020 al estado actual. Contaba con un dataset de viviendas de 2020 obtenido de la web pisos.com, con lo que la API de idealista no me daría los precios de hace un año

**Plan B:** sacar ‘a pedaleta’ los datos mediante un *crawler* montado con la librería BeautifulSoup de Python en un notebook del entorno Jupyter y la librería Geopy para sacar las distancias de las viviendas al centro de Valladolid (Plaza Mayor) y el centro de Bilbao (Casco Antiguo)

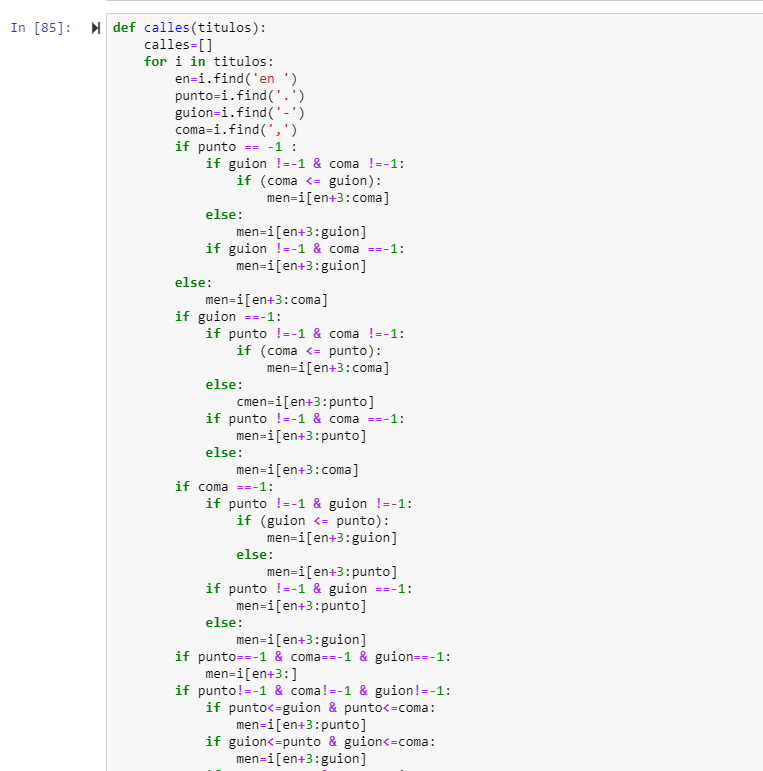
<https://www.pisos.com/venta/pisos-valladolid_capital/>

<https://www.pisos.com/venta/pisos-bilbao/>

Además, me parecía de interés añadir a la tabla anterior la variable *renta\_media\_barrio\_persona*, que es la renta media disponible por persona de cada barrio. Esta información la he encontrado en los dos links siguientes:

<https://www.eustat.eus/elementos/ele0006200/Renta_personal_media_de_la_CA_de_Euskadi_por_barrio_de_residencia_de_las_capitales_segun__tipo_de_renta_euros/tbl0006267_c.html>

<https://www.20minutos.es/noticia/3802324/0/centro-fue-barrio-valladolid-con-mas-renta-disponible-media-2017-rondilla-tuvo-mas-baja/>

Variables más difíciles:**Distancia al centro** –>BeautifulSoup: extracción dirección vivienda (calles) 🡪 Geopy: obtengo latitud, long y transformo





**Dataset salida**



1. **TRANSFORMACION DE DATOS**

Origen: dataset anterior

**PREPROCESAMIENTO**

Los datos sacados de las urls de pisos.com, no salen directamente para entrenar modelos de ML o DL, sino que requieren unos pasos previos de preprocesado y limpieza.

Me interesa conocer de primeras la variable **evol\_precio** (evolucion del precio de las viviendas que aun quedan en el portal inmobiliario desde abril de 2020), para ello, extraigo los datos del dataset del 2020 que ya previamente habia extraído en abril 2020 y los junto (hago un merge inner join) entre ambos datasets. Lo más sencillo hubiese sido unirlos por la referencia del anuncio, pero teniendo en cuenta que esta puede cambiar y que en el dataset del 2020 no dispongo de esta información, hago el merge a través de otras características.





Observo que haciendo el merge, solo me quedo con 1490 entradas, es decir, reduzco a menos de la mitad los datos que tenía, con lo cual, no sé como me va a funcionar al hacer el modelado con tan pocos datos. Por tanto, decido obviar la variable evol\_precio para el modelado de datos, y sí lo utilizaré para el análisis estadístico y la visualización de datos en Tableau (dataset: pisos\_cleaned\_agrupando\_merge\_2021)

Podía haber hecho un right join para no penalizar la cantidad de datos y mostrarme los pisos aun existentes desde 2020…

Al igual que el paso anterior, el limpiar y transformar los datos es fundamental para que el modelo funcione adecuadamente.

Las principales transformaciones que se han hecho han sido:

* Muchos anuncios eran clasificados como apartamento, cuando el número de habitaciones era superior a 1.
* Varios anuncios eran clasificados como áticos pero la planta era un bajo o un entresuelo, o en la descripción, se definía como otro tipo de vivienda, por tanto, he tenido que hacer transformaciones en la variable *tipo\_vivienda* en función de la variable *planta* y viceversa.
* Garaje (trastero/ascensor): generalmente, este dato lo obtengo directamente de las características del anuncio. Aunque no todos los inmuebles con garaje tenían en sus características que disponían de garaje, por tanto, he tenido que recurrir a la descripción para obtener este dato, creando una variable nueva *garaje\_en\_desc* que, me indique si la palabra garaje se encuentra o no en la descripción.

Si se encuentra en descripción, tengo que ver en qué condiciones se encuentra la palabra. Por ejemplo, si aparece como ‘con garaje’, ‘tiene garaje’ me indica que el inmueble tiene garaje, pero si aparece como ‘sin garaje’, ‘no tiene garaje’, me indica que no tiene garaje.

Los mismos pasos realizados con *trastero* y *ascensor.*

***Datos nulos***

*Aunque hay varias formas de tratar los nulos, la que he preferido emplear es directamente la* ***eliminación*** *de ellos, para que el modelo no contenga ningún dato inferido.*



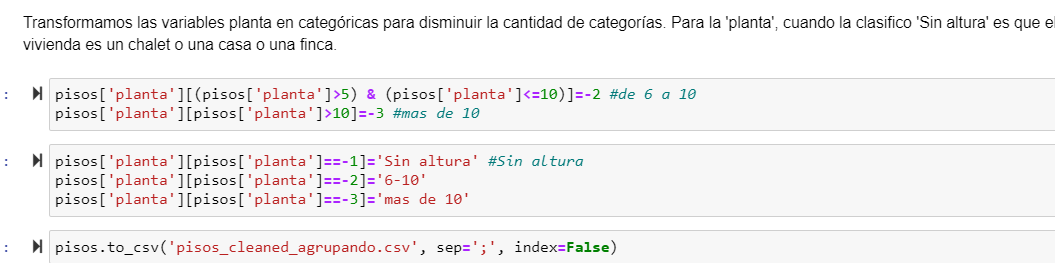
**Manejo de datos categóricos:**

Al manejar datos categóricos, se trabaja con características ordinales y nominales. Las características ordinales son características categóricas que pueden ser ordenadas, mientras que las características nominales no implican ningún orden (tipo\_vivienda).

**¿Influye en el modelo que la planta sea numérica o nominal? Por experiencia: no**

Cuando la variable **Planta es numérica**, toma los valores de **0 a 20**.

Cuando la variable **Planta es nominal**, toma los valores: **Sin altura** (para aquellas casas o chalets donde la altura de la vivienda no la tengo en cuenta), **0, 1 , 2, 3, 4, 5, 6-10, más de 10.**



**Conversión de características ordinales:** para asegurarme de que el algoritmo interpreta de forma correcta las características ordinales, es necesario convertir los valores a números enteros, de forma manual. Por ejemplo, en el caso de planta:

Bajo:0, Planta1:1, Planta2:2, etc. Planta: Sin altura -- solo para casas o chalets unifamiliares

Dado que el objetivo es desarrollar un modelo que tenga la capacidad de predecir el valor de las viviendas, divido el conjunto de datos en las características y en la variable objetivo.

* *Precio:* el precio en euros de cada inmueble anunciado – **VARIABLE OBJETIVO**-
* *titulo:* título del anuncio
* *n\_habitaciones:* Nº de habitaciones (int)
* *n\_banos:* Nº de baños (int)
* *tipo\_vivienda:* si la vivienda corresponde a un piso, apartamento, chalet, casa, ático, dúplex o loft (object)
* *planta:* altura de la vivienda (int en dataset\_sin\_agrupar/object en dataset\_agrupando)
* *garaje:* variable dicotómica que indica si la vivienda tiene o no garaje. No=no tiene garaje, SI=si tiene garaje (int)
* *trastero:* variable dicotómica que indica si la vivienda tiene o no trastero. No=no tiene trastero, SI=si tiene trastero (int)
* *ascensor:* variable dicotómica que indica si la vivienda tiene o no ascensor. No=no tiene ascensor, SI=si tiene ascensor (int)
* *obra\_nueva:* variable dicotómica que indica si la vivienda es de obra nueva o de segunda mano. 1=es de obra nueva, 0=es de segunda mano (int)
* descripción: descripción de la vivienda (object)
* *barrio:* barrio al que pertenece el inmueble anunciado (object)
* *distancia\_centro*
* *lat y long*

**SUPOSICIONES O HIPOTESIS**

Las suposiciones que he generado antes de saber los resultados de mi modelo y las que quiero probar son:

* Las viviendas con más habitaciones (valor de *n\_habitaciones* mayor) tendrán un precio más elevado.
* Las viviendas con mayor altura (valor de *planta* más elevado) tendrán un precio más elevado.
* Las viviendas que estén en un barrio con mayor renta, tendrán un precio más alto.
* Las viviendas que disponen de garaje (valor de *garaje=1*) tendrán un precio más elevado.

**ELIMINOOUTLIERS:**

* Pisos
* Metros
* Longitud y Latitud procedentes de webscraping
* Distancia al centro

**¿Podía haber dado juego a la variable evol\_precio?**

Podría haber hecho un right join entre el dataset de 2021 y el dataset de 2020 para añadir una variable más a mi fuente de datos de 2021 que me permita evaluar la evolución del precio y la influencia en el precio actual de la vivienda.



**Dataset salida:**



1. **DIVISIÓN DE DATOS**

SELECCIONAR CARACTERISTICAS RELEVANTES

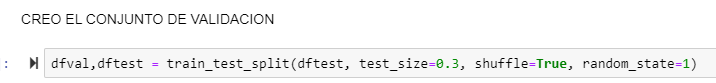
En general, la división de datos se hace en tres partes:

Entreno el modelo con el conjunto de entrenamiento (TRAIN), evalúo el modelo con el conjunto de VALIDACION (VAL) y una vez listo para usarse, lo pruebo con conjunto de PRUEBAS (TEST).

**ENTRENAMIENTO 70%**

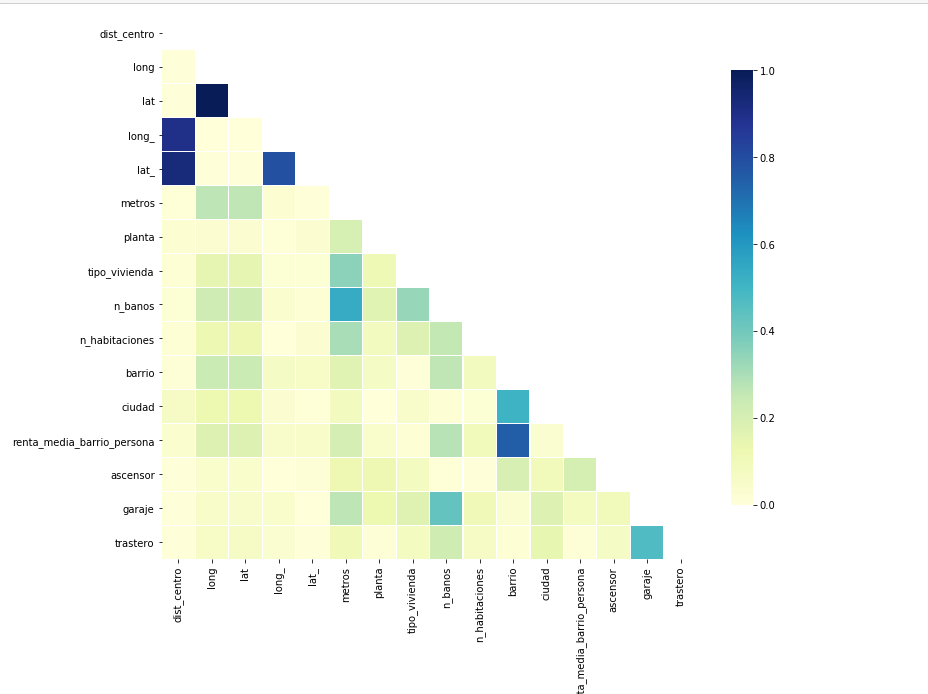
**test 21%**

**Val 9% 🡪 en el fichero Machine Learning**

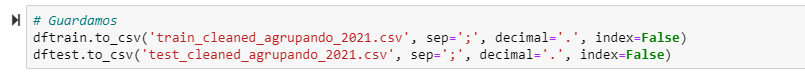


El objetivo último es que el modelo pueda generalizar correctamente con los datos no procesados aún (los de prueba), basándose en los resultados de entrenamiento y prueba.

Matriz de correlaciones



Dataset salida:



1. Machine Learning

**CODIFICACION DE ETIQUETAS DE CLASES NOMINALES**

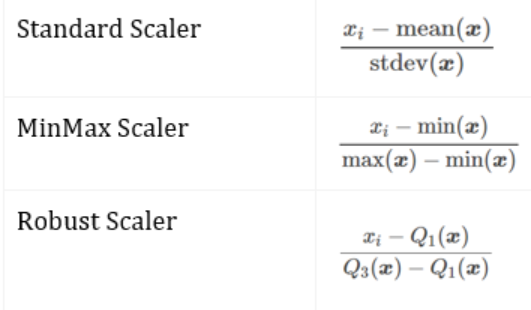
En mi modelo, hay varias características con etiquetas de clases nominales. Por tanto, para que el algoritmo pueda manejar estas etiquetas, tengo que pasarlas a numéricas mediante **mean encoder** (por ejemplo, aplicada en la característica *barrio*).

**ESCALACIÓN DE CARACTERÍSTICAS**

La mayoría de los algoritmos de ML y DL tienen mucho mejor rendimiento cuando trabajan características que están en la misma escala. Y esto es lo que he hecho en todos los algoritmos de ML y DL empleados.

Las técnicas que he aplicado han sido:

* Normalización – **MinMax Scaler**, en Deep Learning
* Estandarización 🡪 **Standard\_Scaler** en Machine Learning



**OVERFITTING Y UNDERFITTING**

Uno de los problemas más importantes cuando se trabaja con el entrenamiento de modelos, es el dilema **optimización** (que afecta al Score Train) versus **generalización** (que afecta al Score Test).

La **optimización** es el proceso de ajuste del modelo para conseguir el mejor rendimiento de los datos de Train (proceso de aprendizaje).(Sub-ajuste / Sobre-ajuste)

La **generalización** es la capacidad del modelo de obtener buenos resultados con otros datos no procesados aún.

EL objetivo que se quiere conseguir con un modelo es que según se vaya generando el proceso de aprendizaje de los datos de entrenamiento, el modelo aprenda bien esos datos pero que no aprenda pautas demasiado concretas de estos datos que puedan ser irrelevantes para los nuevos datos de prueba.

**REGULARIZACION**

El **overfitting o sobre-ajuste**, que hace que el modelo no generalice bien, se puede evitar mediante la regularización. Restringe la complejidad de la red forzando sus pesos a tomar pequeños valores.

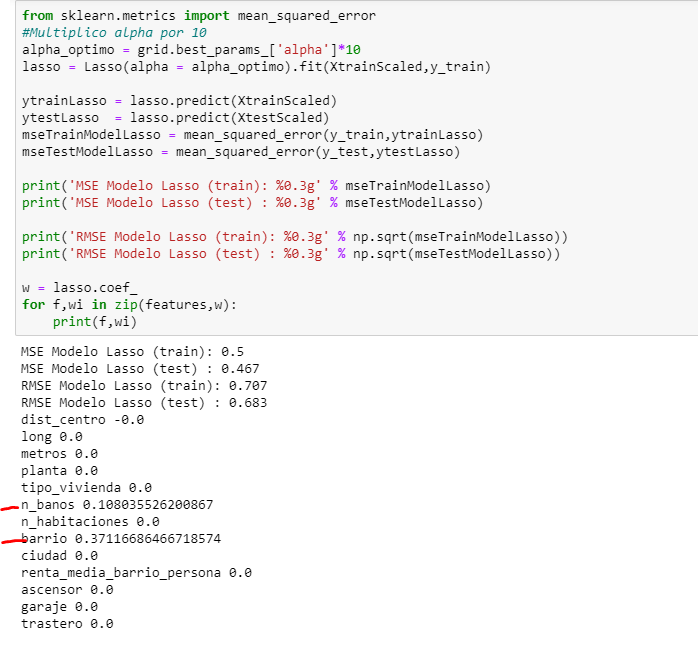
* En la parte de ML, he utilizado la regularización de **LASSO.**

**AJUSTE DE HIPERPARÁMETROS con GSCV**

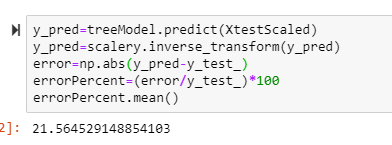
Los hiperparámetros, al contrario que los parámetros que se aprenden en el proceso de aprendizaje del modelo, son los que yo meto algoritmo. En MACHINE LEARNING, para encontrar la mejor combinación de hiperparámetros, utilizo **Grid Search Cross Validation.**

**Utilizo para mi modelo: LASSO + ARBOL DE DECISION + RANDOM FOREST + BOOSTING**

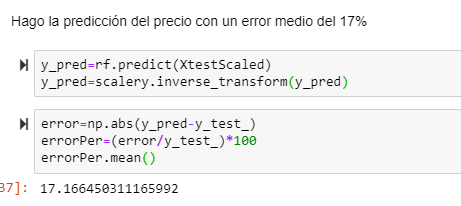
**LASSO: barrio, n\_baños**

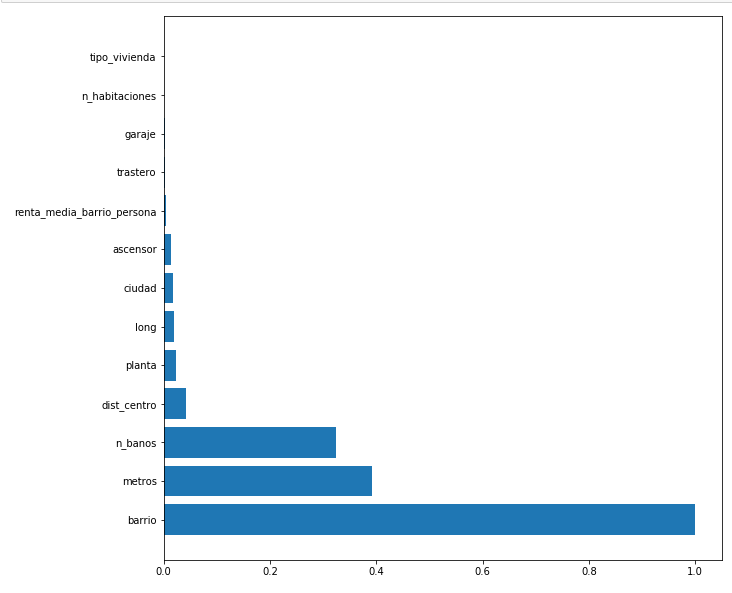


**ARBOL DE DECISION: 21% error**

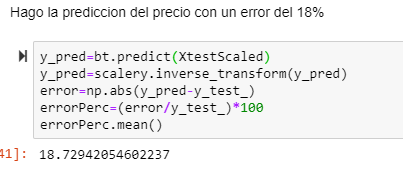


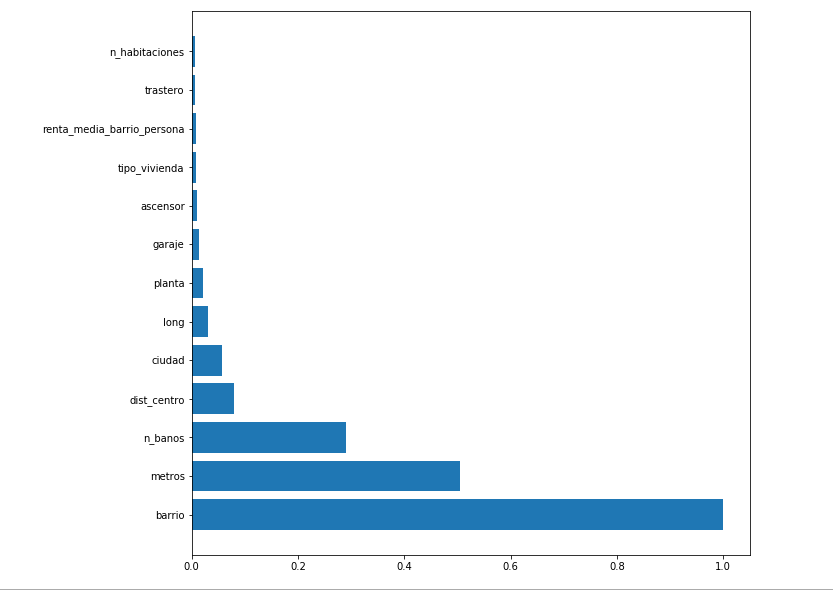
**RANDOM FOREST: 17 % error. Caract más influyentes: Barrio, metros, n\_baños, dist al centro**





**BOOSTING: 18 % error. Características más influyentes en el precio: Barrio, metros, n\_baños**





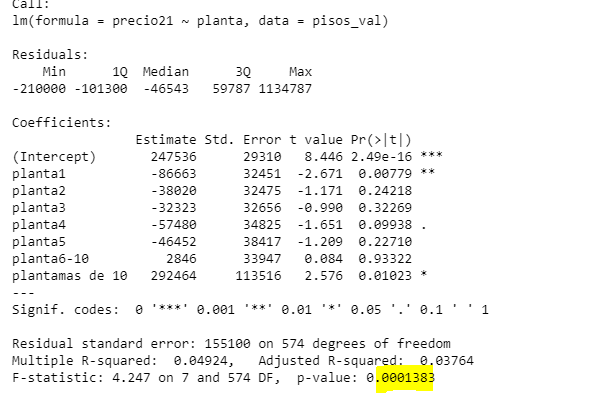
Las caracteristicas mas importantes para el precio son:

**Barrio, metros, baños, distancia al centro, ciudad y planta**

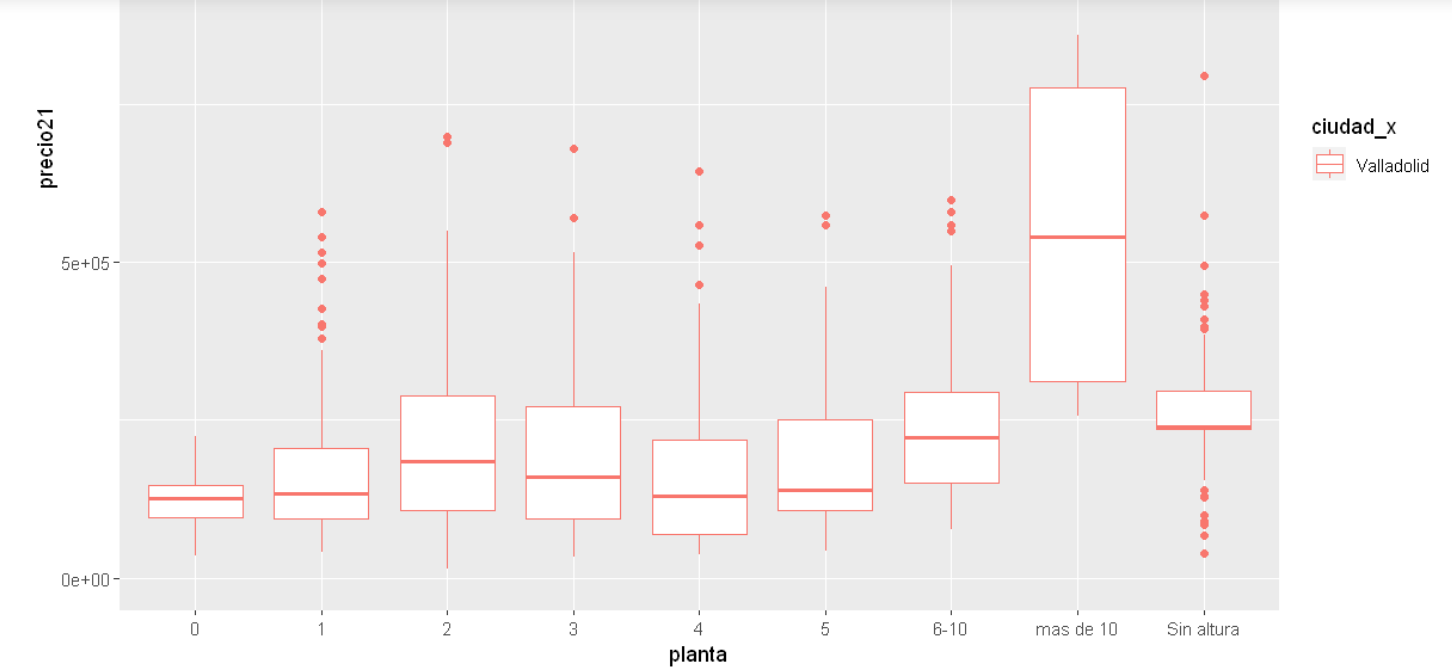
1. **ANÁLISIS ESTADISTICO MEDIANTE R**

Por ultimo, hago un análisis estadístico para ver la influencia de las variables en la predicción

**Influencia de la PLANTA en la predicción del precio de los inmuebles de Valladolid**

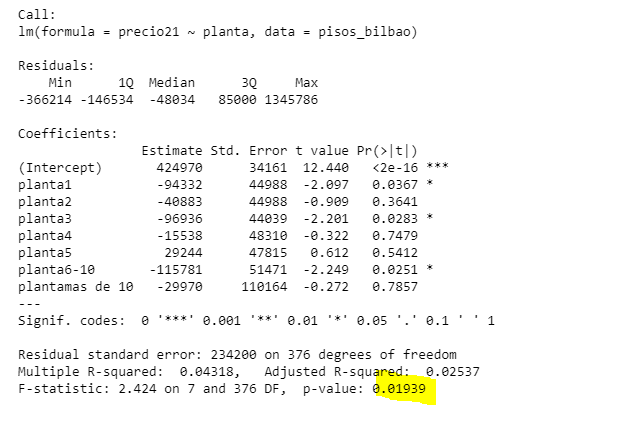
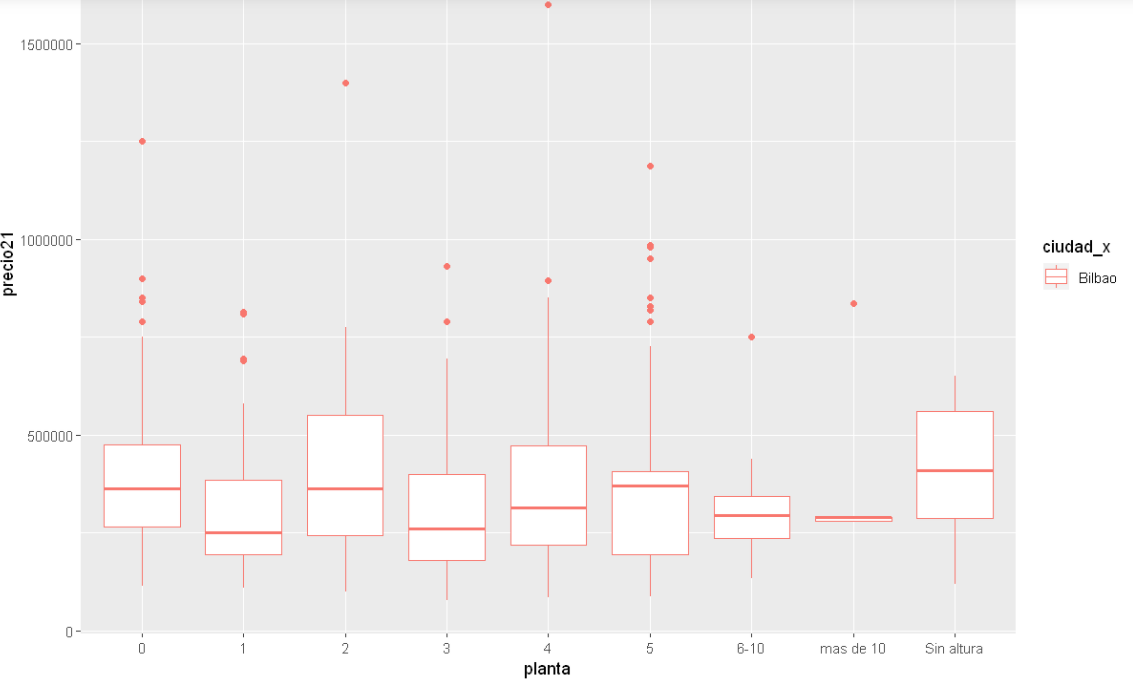


La planta si tiene influencia para los inmuebles de Valladolid a partir de la 6º altura



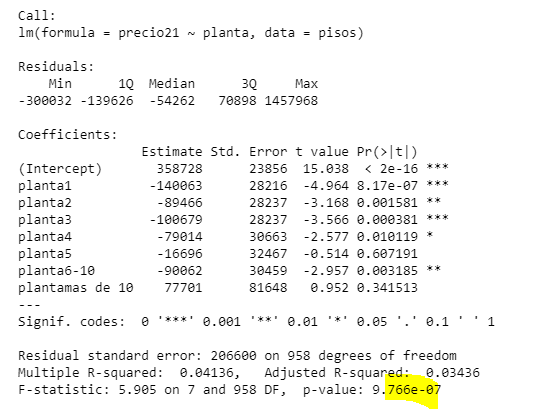
**Influencia de la PLANTA en la predicción del precio de los inmuebles de Bilbao**

En el gráfico inferior lo veo claramente que no hay diferencias entre las medianas en función de cada planta y el pvalor es mayor a 0,05. Por tanto, para Bilbao, la altura no influiría en el precio.

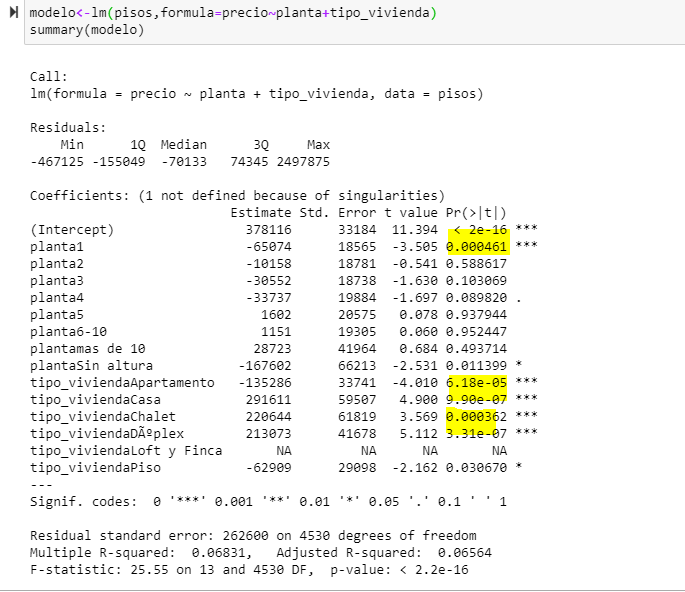
**Influencia de la PLANTA en la predicción del precio de los inmuebles de Bilbao y Valladolid sin sesgar por ciudad**

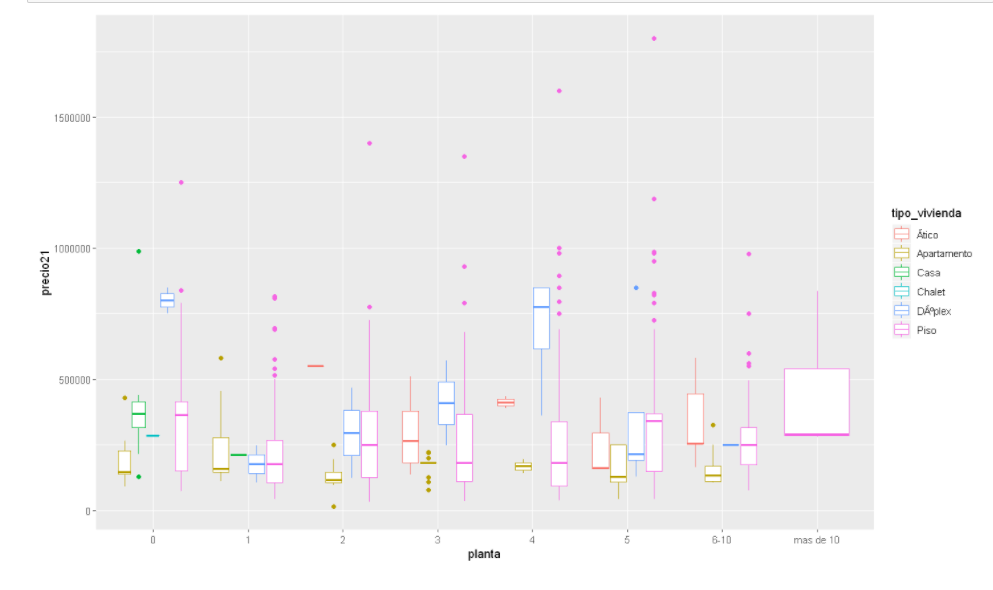
Veo que sin sesgar por ciudad, la planta sí tiene gran influencia en el precio.



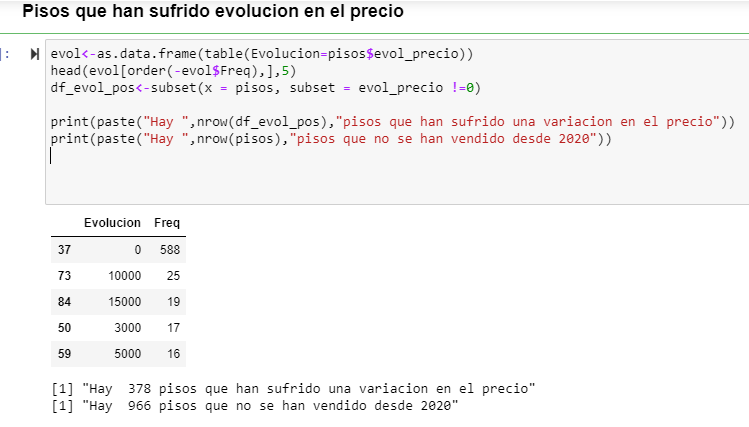
**Influencia de la interacción de la planta por tipo de vivienda**

Veo que para piso, el precio no dependería de la altura.

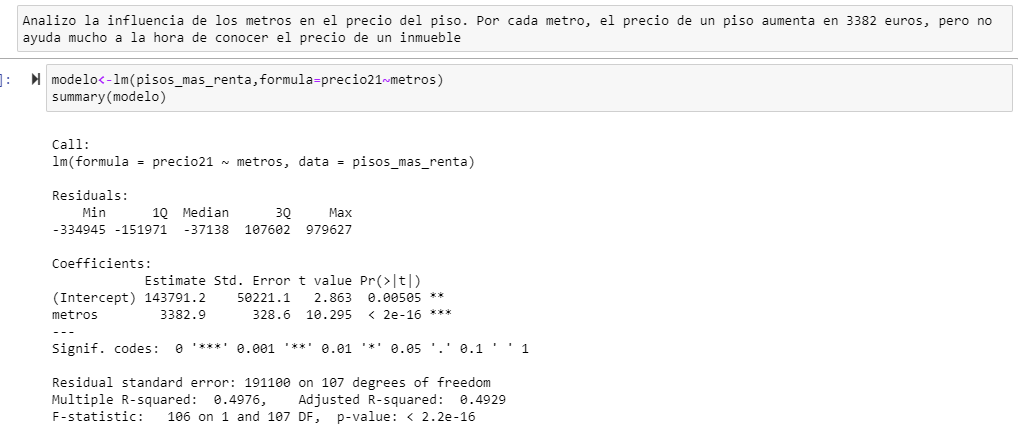




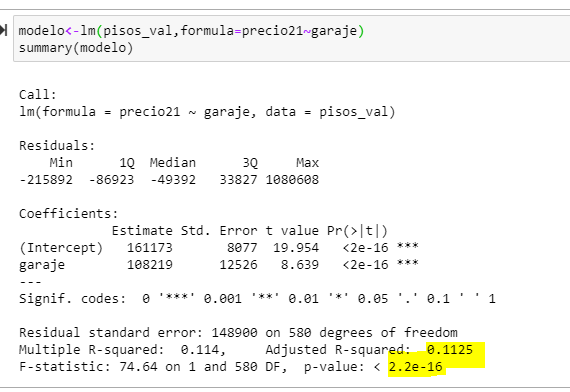
**Evolución en precio**



**Cuánto aumenta el precio cada metro cuadrado de más: 3382 €uros en general**



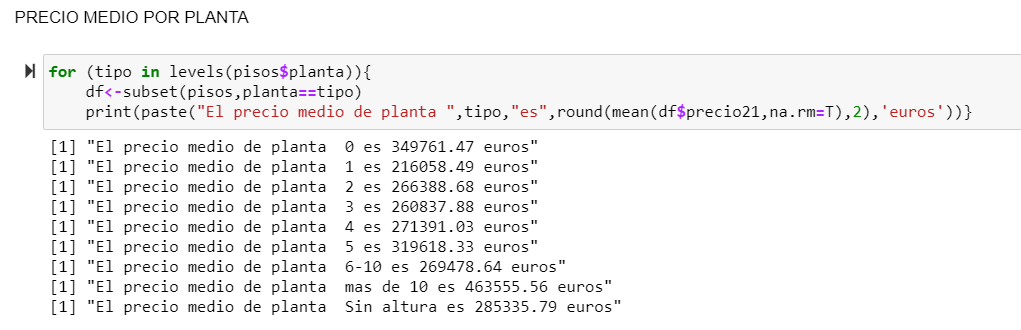
**¿Cuánto aumenta el precio en Valladolid por tener garaje? 108219 €,** aunque no nos ayuda mucho la variable garaje a predecir el precio (R-squared bajo)



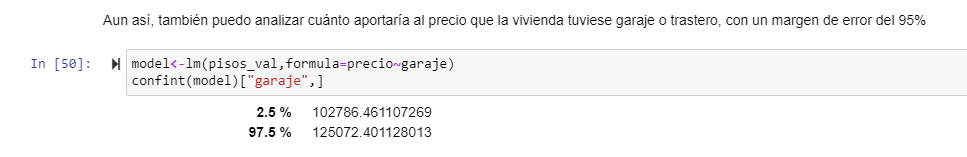
**Precio medio por ciudad**



**Precio medio por planta**

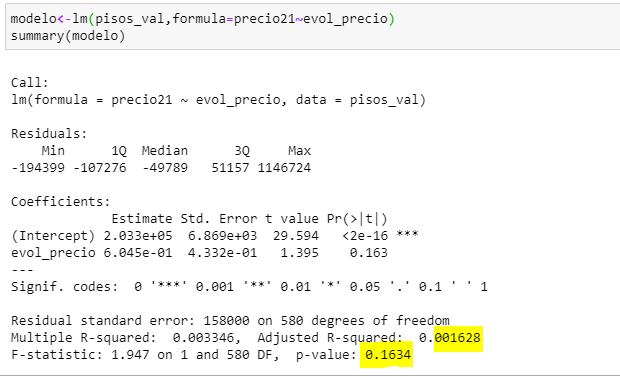


**Influencia en el precio de garaje en los inmuebles de Valladolid**

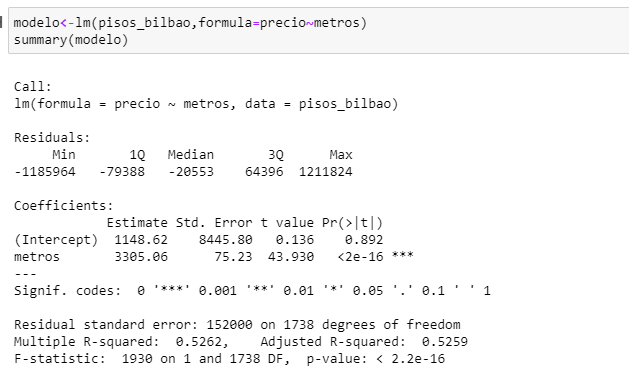


**Influencia de la evolución del precio en los inmuebles de Valladolid**

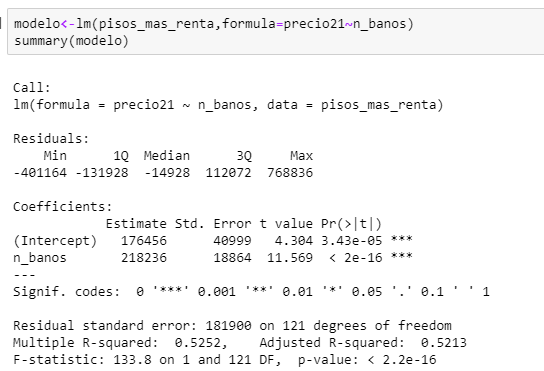
Vemos que no tiene influencia (p\_valor es grande >0.05) y tampoco nos ayuda a predecir por sí solo el precio del inmueble (Rsquared muy pequeño)



**Cuanto aumenta el precio de un inmueble en Bilbao por cada metro cuadrado de más: 3305€**



**Cuánto aumenta el precio de un inmueble por cada baño adicional? 21**



Y hallando el conjunto de variables que mejor nos ayudarían a predecir el precio:

* con un R-squared de 0.827 (también lo puedo calcular mediante data mining con un GLM)

modelo<-lm(pisos,formula=precio21~n\_banos+metros+barrio+evol\_precio+garaje)

FINALIZO RESOLVIENDO LAS HIPOTESIS INICIALES

SUPOSICIONES O HIPOTESIS

Las suposiciones que he generado antes de saber los resultados de mi modelo y las que quiero probar son:

* Las viviendas con mayor altura (valor de *planta* más elevado) tendrán un precio más elevado. **Por lo general no, si nos enfocamos en tipo de vivienda y ciudad, por ejemplo, los pisos de Valladolid, el precio crece con la altura. Para Bilbao, por el contrario, no.**
* Las viviendas que estén en un barrio con mayor renta, tendrán un precio más alto. **NO**
* Las viviendas que disponen de garaje (valor de *garaje=1*) tendrán un precio más elevado. **SI**

1. **Visualizacion de datos TABLEAU**

**Gráficos:**

* **Tipo vivienda:** nº de inmuebles de la clase seleccionada
* **Número de registros por ciudad:** cantidad de inmuebles por ciudad, y cuántos tienen garaje
* **Precio medio por metros:** en general, para la mayoría de casos, se puede decir que a mayor tamaño, mayor precio
* **Precio medio por planta y vivienda:**
* **Evolución precio por ciudad:** nº de inmuebles de la clase seleccionada que han sufrido una variación de precio desde abril de 2020 (positiva, si ha bajado de precio; negativa, si ha subido)
* **Precio medio vs distancia al centro:** precio medio en orden ascendente de los inmuebles de la clase seleccionada y de la ciudad seleccionada por distancia al centro (en km) (Bilbao: Casco Viejo; Valladolid: Plaza Mayor)🡪 podemos ver que la distancia al centro de la ciudad no es significativa del precio de la vivienda.

