Web scraping

In [1]:
```python
import bs4 as bs
import urllib
import urllib.request
import re
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import io
import urllib.robotparser
```

Dondes una ciutat i un html retorna la taula comparativa que es troba a la web. Retorna una taula amb els valors següents:

Població

Valor de l'immoble segons: Menoss 60 m2 / Menos de 100.000€

Valor de l'immoble segons: Menos 60 m2 / Entre 100.000 € y 250.000 €

Valor de l'immoble segons: Menos 60 m2 / Más de 250.000€

Valor de l'immoble segons: Entre 60 m2 y 120 m2 / Menos de 100.000€

Valor de l'immoble segons: Entre 60 m2 y 120 m2 / Entre 100.000 € y 250.000 €

Valor de l'immoble segons: Entre 60 m2 y 120 m2 / Más de 250.000€

Valor de l'immoble segons: Más de 120 m2 / Menos de 100.000€

Valor de l'immoble segons: Más de 120 m2 / Entre 100.000 € y 250.000 €

Valor de l'immoble segons: Más de 120 m2 / Más de 250.000€

In [2]:
```python
def obtenirComparativa(ciutat,HTML):
    historypage = urllib.request.urlopen(HTML)
    soup = bs.BeautifulSoup(historypage,'html.parser')

    #Les dades es troben en un div amb la classe locality-stats__table hidden-xs
    divComparativa = soup.find('div', {'class':'locality-stats__table hidden-xs'})

    #Recorrem els div i a la classe rate-value trobem el valor que busquem. Ho guardem tot a la taula tablaComp
    tablacomp= []
    tablacomp.append(ciutat)
    for rates in divComparativa.findAll ('div',{'class':'rate-value'}):
        tablacomp.append(rates.find(text=True))

    return (tablacomp)
```

In [3]:
```python
def generarCSV(taula,nomFitxer):
    taula.to_csv(nomFitxer + '.csv', header=True, index = False)
```

In [4]:
```python
HTML = "https://www.trovimap.com/precio-vivienda/"
```

mirem a robots.txt per veure si hi ha pàgines bloquejades a robots

In [5]:
```python
rp = urllib.robotparser.RobotFileParser()
rp.set_url('https://www.trovimap.com/robots.txt')
rp.read()
user_agent= '*'
```

In [6]:
```python
pais = "espana"
HTMLEspana = HTML + pais

if rp.can_fetch(user_agent, HTMLEspana):
    # Amb .read_html() podem obtenir fàcilment la taula que ens mostra la web.
    df = pd.read_html(HTMLEspana)
    dfProvinciesEspana = df[0]
    dfProvinciesEspana = dfProvinciesEspana.rename(columns={"Unnamed: 0": 'Provincies'})
else:
    print ("Pàgina bloquejada per robots,txt: " + HTMLEspana)
```

In [7]:
```python
dfProvinciesEspana.head()
```

Out[7]:

	Provincies	Variación Mensual	Variación 3 meses	Variación anual	€/m2
0	A Coruña	-0.91%	+0.01%	-6.62%	1.476 €
1	Albacete	+1.42%	+5.63%	-1.38%	1.280 €
2	Alicante/Alacant	+0.31%	+0.07%	-0.36%	1.727 €
3	Almería	+2.48%	+6.13%	+4.27%	1.223 €
4	Araba/Álava	-1.96%	-1.71%	-6.03%	2.051 €

In [8]:
```python
dfProvinciesEspana.iloc[0]['Provincies']
```

Out[8]:
```
'A Coruña'
```

Obtenim una taula amb la variació menusal de cada provincia de l'estat.

Si volem obtenir una taula, tenint per files les ciutats més importants de cada provincia tal com hem fet anteriorment amb la de Barcelona:

Si volem obtenir una taula, tenint per files les ciutats més importants de Catalunta:

In [9]:
```python
# Agafar la taula per cada una de les quatre provincies de Catalunya.
dfBarcelona = pd.read_html(HTML+"barcelona")
dfTarragona = pd.read_html(HTML+"tarragona")
dfGirona = pd.read_html(HTML+"girona")
dfLleida = pd.read_html(HTML+"lleida")
dfBarcelona = dfBarcelona[0]
dfTarragona = dfTarragona[0]
dfGirona = dfGirona[0]
dfLleida = dfLleida[0]

# Concatenar Taules.
dfCatalunya = pd.concat([dfBarcelona, dfTarragona], ignore_index=True)
dfCatalunya = pd.concat([dfCatalunya, dfGirona], ignore_index=True)
dfCatalunya = pd.concat([dfCatalunya, dfLleida], ignore_index=True)

# Renombrar columnes.
dfCatalunya = dfCatalunya.rename(columns={"Unnamed: 0": 'Ciutat', "Variación Mensual": 'Variació mensual',
                                "Variación 3 meses": 'Variació tres messos', "Variación anual": 'Variació anual'})

dfCatalunya.tail()
```

Out[9]:

	Ciutat	Variació mensual	Variació tres messos	Variació anual	€/m2
106	Vidreres	-0.93%	-1.03%	-5.30%	1.409 €
107	Alpicat	-	+0.65%	-	1.083 €
108	Lleida	+0.96%	+3.13%	+4.94%	1.210 €
109	Naut Aran	-1.46%	+0.83%	-	3.891 €
110	Vielha e Mijaran	-2.10%	+6.50%	-1.93%	2.331 €

Ara volem treure totes les ciutats importants de España. Per fer això necesitem totes les url de cada provincia a partir de el html.

In [10]:
```python
historypage = urllib.request.urlopen(HTMLEspana)
soup = bs.BeautifulSoup(historypage,'html.parser')
makeitastring = ''.join(map(str, soup))
#soup
#makeitastring

# Buscar el patró dins del html on es guarda els noms de les provincies per despres buscar-les en url.
capitals = '/'.join(map(str, re.findall("precio-vivienda(.+?)\">",makeitastring)))
capitals = capitals[46:570]
print(capitals)

# Ja tenim un array amb tots els noms de les provincies que haurem de posar a la url.
capitalsArray = capitals.split('//')
capitalsArray
```

```
a-coruna//albacete//alicante-alacant//almeria//araba-alava//asturias//avila//badajoz//barcelona//bizkaia//burgos//caceres//cadi
z//cantabria//castellon-castello//ceuta//ciudad-real//cordoba//cuenca//gipuzkoa//girona//granada//guadalajara//huelva//huesca//
illes-balears//jaen//la-rioja//las-palmas//leon//lleida//lugo//madrid//malaga//melilla//murcia//navarra//ourense//palencia//pon
tevedra//salamanca//santa-cruz-de-tenerife//segovia//sevilla//soria//tarragona//teruel//toledo//valencia-valencia//valladolid//
zamora//zaragoza
```

Out[10]:
```
['a-coruna',
 'albacete',
 'alicante-alacant',
 'almeria',
 'araba-alava',
 'asturias',
 'avila',
 'badajoz',
 'barcelona',
 'bizkaia',
 'burgos',
 'caceres',
 'cadiz',
 'cantabria',
 'castellon-castello',
 'ceuta',
 'ciudad-real',
 'cordoba',
 'cuenca',
 'gipuzkoa',
 'girona',
 'granada',
 'guadalajara',
 'huelva',
 'huesca',
 'illes-balears',
 'jaen',
 'la-rioja',
 'las-palmas',
 'leon',
 'lleida',
 'lugo',
 'madrid',
 'malaga',
 'melilla',
 'murcia',
 'navarra',
 'ourense',
 'palencia',
 'pontevedra',
 'salamanca',
 'santa-cruz-de-tenerife',
 'segovia',
 'sevilla',
 'soria',
 'tarragona',
 'teruel',
 'toledo',
 'valencia-valencia',
 'valladolid',
 'zamora',
 'zaragoza']
```

Ara volem treure totes les ciutats importants de España. Per fer això necesitem totes les url de cada provincia a partir de el html.

In [11]:
```python
historypage = urllib.request.urlopen(HTMLEspana)
soup = bs.BeautifulSoup(historypage,'html.parser')
makeitastring = ''.join(map(str, soup))
#soup
#makeitastring

# Buscar el patró dins del html on es guarda els noms de les provincies per despres buscar-les en url.
capitals = '/'.join(map(str, re.findall("precio-vivienda(.+?)\">",makeitastring)))
capitals = capitals[46:570]
print(capitals)

# Ja tenim un array amb tots els noms de les provincies que haurem de posar a la url.
capitalsArray = capitals.split('//')
capitalsArray
```

```
a-coruna//albacete//alicante-alacant//almeria//araba-alava//asturias//avila//badajoz//barcelona//bizkaia//burgos//caceres//cadi
z//cantabria//castellon-castello//ceuta//ciudad-real//cordoba//cuenca//gipuzkoa//girona//granada//guadalajara//huelva//huesca//
illes-balears//jaen//la-rioja//las-palmas//leon//lleida//lugo//madrid//malaga//melilla//murcia//navarra//ourense//palencia//pon
tevedra//salamanca//santa-cruz-de-tenerife//segovia//sevilla//soria//tarragona//teruel//toledo//valencia-valencia//valladolid//
zamora//zaragoza
```

Out[11]:
```
['a-coruna',
 'albacete',
 'alicante-alacant',
 'almeria',
 'araba-alava',
 'asturias',
 'avila',
 'badajoz',
 'barcelona',
 'bizkaia',
 'burgos',
 'caceres',
 'cadiz',
 'cantabria',
 'castellon-castello',
 'ceuta',
 'ciudad-real',
 'cordoba',
 'cuenca',
 'gipuzkoa',
 'girona',
 'granada',
 'guadalajara',
 'huelva',
 'huesca',
 'illes-balears',
 'jaen',
 'la-rioja',
 'las-palmas',
 'leon',
 'lleida',
 'lugo',
 'madrid',
 'malaga',
 'melilla',
 'murcia',
 'navarra',
 'ourense',
 'palencia',
 'pontevedra',
 'salamanca',
 'santa-cruz-de-tenerife',
 'segovia',
 'sevilla',
 'soria',
 'tarragona',
 'teruel',
 'toledo',
 'valencia-valencia',
 'valladolid',
 'zamora',
 'zaragoza']
```

In [12]:
```python
# Provem amb una provincia
HTML = "https://www.trovimap.com/precio-vivienda/"
HTML + capitalsArray[25]

df = pd.read_html(HTML + capitalsArray[25])
df[0].head()
```

Out[12]:

	Unnamed: 0	Variación Mensual	Variación 3 meses	Variación anual	€/m2
0	Alaior	-0.67%	-	+10.07%	2.130 €
1	Alcúdia	+0.22%	-5.53%	-1.42%	2.784 €
2	Andratx	+3.61%	+0.85%	+5.89%	2.944 €
3	Calvià	+0.29%	+0.05%	+5.34%	3.769 €
4	Campos	-2.92%	-2.34%	+17.20%	2.449 €

In [13]:
```python
# Veiem que serveix, ara només hem de posar-lo en un loop per obtenir totes les ciutats importants d'España.
dfEspana = pd.DataFrame()
tPoblacions = []

for capital in capitalsArray:
    if capital != "ceuta" and capital != "soria":
     #if capital == "tarragona":
        # Per a cada url agafar la taula
        if rp.can_fetch(user_agent, HTML+capital):

            dfAux = pd.read_html(HTML+capital)
            dfAux = dfAux[0]

            # Busquem el nom del poble amb el que es forma el link per buscar a dins la taula comparativa

            poblacionsPage = urllib.request.urlopen(HTML+capital)
            soup2 = bs.BeautifulSoup(poblacionsPage,'html.parser')
            makeitastring = ''.join(map(str, soup2))

            poblacions = '/'.join(map(str, re.findall("precio-vivienda(.+?)\">",makeitastring)))
            poblacionsArray = poblacions.split('//')

            # La primera posició és un títol i la segona posició és la provincia per tant els eliminem
            # La última posició és un link a estadístiques. També l'eliminem.
            poblacionsArray.pop(0)
            poblacionsArray.pop(0)
            poblacionsArray.pop(len(poblacionsArray)-1)

    # Busquem per cada poblacio de cada capital els valors de la taula comparativa
            TP1 = []
            TP2 = []
            TP3 = []
            TP4 = []
            TP5 = []
            TP6 = []
            TP7 = []
            TP8 = []
            TP9 = []
            for pobl in poblacionsArray:
                    tComp = obtenirComparativa (pobl,HTML+pobl)
                    TP1.append(tComp[1])
                    TP2.append(tComp[2])
                    TP3.append(tComp[3])
                    TP4.append(tComp[4])
                    TP5.append(tComp[5])
                    TP6.append(tComp[6])
                    TP7.append(tComp[7])
                    TP8.append(tComp[8])
                    TP9.append(tComp[9])

            dfAux = dfAux.assign (T1=TP1)
            dfAux = dfAux.assign (T2=TP2)
            dfAux = dfAux.assign (T3=TP3)
            dfAux = dfAux.assign (T4=TP4)
            dfAux = dfAux.assign (T5=TP5)
            dfAux = dfAux.assign (T6=TP6)
            dfAux = dfAux.assign (T7=TP7)
            dfAux = dfAux.assign (T8=TP8)
            dfAux = dfAux.assign (T9=TP9)


            #Anar concatenant ciutats
            dfEspana = pd.concat([dfEspana, dfAux], ignore_index=True)
            #print(capital)
        else:
            print ("Pàgina bloquejada per robots,txt: " + HTML+capital)
```

In [14]:
```python
# Renombrar columnes.
dfEspana = dfEspana.rename(columns={"Unnamed: 0": 'Ciutat', "Variación Mensual": 'Variació mensual',
                                "Variación 3 meses": 'Variació tres messos', "Variación anual": 'Variació anual',
                                "T1":'Menos 60 m2 / Menos de 100.000 Euros',"T2":'Menos 60 m2 / Entre 100.000 Euros y 250.000 Eu
ros',
                                "T3":'Menos 60 m2 / Más de 250.000 euros',
                                "T4":'Entre 60 m2 y 120 m2 / Menos de 100.000 Euros',
                                "T5":'Entre 60 m2 y 120 m2 / Entre 100.000 Euros y 250.000 Euros',
                                "T6":'Entre 60 m2 y 120 m2 / Más de 250.000 Euros',
                                "T7":'Más de 120 m2 / Menos de 100.000 euros',
                                "T8":'Más de 120 m2 / Entre 100.000 euros y 250.000 euros',
                                "T9":'Más de 120 m2 / Más de 250.000 euros'
                                })

dfEspana.tail()
```

Out[14]:

	Ciutat	Variació mensual	Variació tres messos	Variació anual	€/m2	Menos 60 m2 / Menos de 100.000 Euros	Menos 60 m2 / Entre 100.000 Euros y 250.000 Euros	Menos 60 m2 / Más de 250.000 euros	Entre 60 m2 y 120 m2 / Menos de 100.000 Euros	Entre 60 m2 y 120 m2 / Entre 100.000 Euros y 250.000 Euros	Entre 60 m2 y 120 m2 / Más de 250.000 Euros	Más de 120 m2 / Menos de 100.000 euros	Más de 120 m2 / Entre 100.000 euros y 250.000 euros	Más de 120 m2 / Más de 250.000 euros
630	Zamora	+0.65%	+2.24%	+1.27%	1.284 €	1.120 €	-	-	909 €	1.579 €	-	605 €	1.123 €	1.560 €
631	Calatayud	+2.31%	-	+6.26%	882 €	-	-	-	764 €	1.271 €	-	-	-	-
632	Cuarte de Huerva	+2.42%	+3.17%	-6.14%	1.352 €	-	-	-	1.035 €	1.685 €	-	-	978 €	-
633	Utebo	-	-6.41%	-	1.363 €	-	-	-	1.060 €	1.619 €	-	-	1.177 €	-
634	Zaragoza	-0.47%	-0.85%	+1.00%	1.684 €	1.340 €	2.431 €	-	1.085 €	1.880 €	3.037 €	-	1.155 €	1.992 €

Ja tenim taules amb les variacions econòmiques de les ciutats més importants tant de Catalunya com d'España.

Una mica de preprocessing

In [30]:
```python
dfEspana['Variació mensual'] = dfEspana['Variació mensual'].str.replace('+','')
dfEspana['Variació tres messos'] = dfEspana['Variació tres messos'].str.replace('+','')
dfEspana['Variació anual'] = dfEspana['Variació anual'].str.replace('+','')
dfEspana['Variació mensual'] = dfEspana['Variació mensual'].str.replace('%','')
dfEspana['Variació tres messos'] = dfEspana['Variació tres messos'].str.replace('%','')
dfEspana['Variació anual'] = dfEspana['Variació anual'].str.replace('%','')

dfEspana['€/m2'] = dfEspana['€/m2'].str.replace('€','')
dfEspana['Menos 60 m2 / Menos de 100.000 Euros'] = dfEspana['Menos 60 m2 / Menos de 100.000 Euros'].str.replace('€','')
dfEspana['Menos 60 m2 / Entre 100.000 Euros y 250.000 Euros'] = dfEspana['Menos 60 m2 / Entre 100.000 Euros y 250.000 Euros'].st
r.replace('€','')
dfEspana['Menos 60 m2 / Más de 250.000 euros'] = dfEspana['Menos 60 m2 / Más de 250.000 euros'].str.replace('€','')
dfEspana['Entre 60 m2 y 120 m2 / Menos de 100.000 Euros'] = dfEspana['Entre 60 m2 y 120 m2 / Menos de 100.000 Euros'].str.replac
e('€','')
dfEspana['Entre 60 m2 y 120 m2 / Entre 100.000 Euros y 250.000 Euros'] = dfEspana['Entre 60 m2 y 120 m2 / Entre 100.000 Euros y
250.000 Euros'].str.replace('€','')
dfEspana['Entre 60 m2 y 120 m2 / Más de 250.000 Euros'] = dfEspana['Entre 60 m2 y 120 m2 / Más de 250.000 Euros'].str.replace
('€','')
dfEspana['Más de 120 m2 / Menos de 100.000 euros'] = dfEspana['Más de 120 m2 / Menos de 100.000 euros'].str.replace('€','')
dfEspana['Más de 120 m2 / Entre 100.000 euros y 250.000 euros'] = dfEspana['Más de 120 m2 / Entre 100.000 euros y 250.000 euro
s'].str.replace('€','')
dfEspana['Más de 120 m2 / Más de 250.000 euros'] = dfEspana['Más de 120 m2 / Más de 250.000 euros'].str.replace('€','')

dfEspana.tail()
```

Out[30]:

	Ciutat	Variació mensual	Variació tres messos	Variació anual	€/m2	Menos 60 m2 / Menos de 100.000 Euros	Menos 60 m2 / Entre 100.000 Euros y 250.000 Euros	Menos 60 m2 / Más de 250.000 euros	Entre 60 m2 y 120 m2 / Menos de 100.000 Euros	Entre 60 m2 y 120 m2 / Entre 100.000 Euros y 250.000 Euros	Entre 60 m2 y 120 m2 / Más de 250.000 Euros	Más de 120 m2 / Menos de 100.000 euros	Más de 120 m2 / Entre 100.000 euros y 250.000 euros	Más de 120 m2 / Más de 250.000 euros
630	Zamora	0.65	2.24	1.27	1.284	1.120	-	-	909	1.579	-	605	1.123	1.560
631	Calatayud	2.31	-	6.26	882	-	-	-	764	1.271	-	-	-	-
632	Cuarte de Huerva	2.42	3.17	-6.14	1.352	-	-	-	1.035	1.685	-	-	978	-
633	Utebo	-	-6.41	-	1.363	-	-	-	1.060	1.619	-	-	1.177	-
634	Zaragoza	-0.47	-0.85	1.00	1.684	1.340	2.431	-	1.085	1.880	3.037	-	1.155	1.992

Volem buscar l'ultima data que es van recollir dades

In [31]:
```python
# Buscar el patró dins del html
historypage = urllib.request.urlopen(HTML+"barcelona")
soup = bs.BeautifulSoup(historypage,'html.parser')
makeitastring = ''.join(map(str, soup))
#makeitastring
calendar = '/'.join(map(str, re.findall("en el mes de (.+?) es ",makeitastring)))
calendar = calendar[:int(len(calendar)/2)]
calendar = calendar.replace(' ', '_')
calendar = '_' + calendar
calendar
```

Out[31]:
```
'_marzo_2020'
```

Gravem a fixter csv totes les poblacions d'espanya amb les seves variacions de preu

In [32]:
```python
generarCSV (dfEspana,'PreusVivendaEspanyaMes'+calendar)
```

Gravem també un altre csv amb les variacions generals de cada província d'espanya

In [33]:
```python
generarCSV(dfProvinciesEspana,'Provincies'+calendar)
```

In [34]:
```python
print("Fi scrapping")
```

```
Fi scrapping
```