Real estates is Czech republic

About this project

My initial idea was to try to scrape any website and clean the acquired data. For this project I chose **sreality.cz**.
I scraped 100 pages and gained basic information about each ad - whether the property is for rent or sale, what kind of property it is, what is the area of the property as well as the land, layout (if the property is a flat), address and price.

Methodology

1. Scraping

The problem that I encountered during scraping was that two different values, that I wanted to scrape, had the same class name. I resolved it using *css :nth-of-type* selector.
Here is part of the scraping code *(the rest can be seen here: https://github.com/ZuzanL/data_projects/blob/main/web_scraping/scrape_realestate.py).* In the next step I created a dictionary which I saved as a json file.

```
# Find all properties and their information on the page
    properties = soup.find_all('div', class_='css-173t8lh')

    #Break the loop if no more products are found
    if not properties:
        break

    for product in properties:
        name = product.find('p', class_='css-d7upve')
        if name:
            p_name = name.get_text()
        else:
            p_name = "N/A"

        address = product.select_one('p.css-d7upve:nth-of-type(2)') # Selects
the second p with class css-d7upve
        if address:
            p_address = address.get_text(strip=True)
        else:
            p_address = "N/A"
```

```
        price = product.find('p', class_='css-ca9wwd')
        if price:
            p_price = price.get_text()
        else:
            p_price = "N/A"
```

2. Cleaning

In this picture is shown how my dataframe loaded from the json file initially looked like.

	name	adress	price
0	Pronájem bytu 2+1 60 m²	Pod Lesem, Plzeň - Doubravka	15 500 Kč/měsíc
1	Prodej bytu 1+1 35 m²	Úvalská, Karlovy Vary - Drahovice	2 600 000 Kč
2	Prodej chalupy 110 m², pozemek 331 m²	Sopotnice	1 890 000 Kč
3	Prodej rodinného domu 110 m², pozemek 331 m²	Sopotnice	1 890 000 Kč
4	Prodej rodinného domu 250 m², pozemek 800 m²	Jičín	5 890 000 Kč
...	...	...	...
2195	Prodej bytu 3+kk 86 m²	Srní	9 890 000 Kč
2196	Prodej bytu 4+kk 125 m²	Karlovarská, Unhošť	7 999 000 Kč
2197	Prodej bytu 1+1 38 m²	Košická, Praha - Vršovice	5 999 999 Kč
2198	Prodej bytu 3+1 81 m²	Rostovská, Praha - Vršovice	14 857 931 Kč
2199	Prodej bytu 1+kk 24 m²	Aloise Rašína, Olomouc - Řepčín	3 250 000 Kč

2200 rows × 3 columns

Biggest challenge, during cleaning this dataset, was extracting data from the first column (**'name'**). My aim was to create new columns:

- Whether the property is on sale or for rent
- What kind of property it is (house, flat, land…)
- What is the area of the property
- What is the layout of the flat (only flats had this information about them)
- What is the area of the land (for every kind of property except the flat)

From column **'address'** I needed to extract the city. Properties in smaller cities or in villages did not have any information about street or city-part, so those were simple (extract just the one word that represented the city). Other properties had information about which street are they on, followed by the city and sometimes also a city part. To include all of the conditions was a bit challenging for me, but this is how I've done it:

```
def extract_city(adress):
    adress_line = adress.split()
    if len(adress_line) == 1:
```

```
        return adress, None  # Single word: village, no city part
    else:
        city_match = re.search(r"([^-,\n]+)(?:\s*-\s*([^,\n]+))?$", adress)
        if city_match:
            city = city_match.group(1).strip() #remove spaces
            city_part = city_match.group(2).strip() if city_match.group(2) else
None #remove spaces
            return city, city_part
        else:
            return None, None  # No match, return None for both

df_copy[['city', 'city_part']] = df_copy['adress'].apply(lambda x:
pd.Series(extract_city(x)))
```

From the last column **'price'** I extracted just the numbers.
The whole code and all the regular expressions I used can be seen here:
https://github.com/ZuzanL/data_projects/blob/main/web_scraping/cleanup_RE_scraped_data%20%E2%80%93%20k%C3%B3pia.ipynb
This is the final data frame that I saved as a csv file.

	name	adress	price	type	RE	area_m2	land_area_m2	disposition	city	city_part	price_KC
0	Pronájem bytu 2+1 60 m²	Pod Lesem, Plzeň - Doubravka	15 500 Kč/měsíc	pronájem	bytu	60	None	2+1	Plzeň	Doubravka	15500.0
1	Prodej bytu 1+1 35 m²	Úvalská, Karlovy Vary - Drahovice	2 600 000 Kč	prodej	bytu	35	None	1+1	Karlovy Vary	Drahovice	2600000.0
2	Prodej chalupy 110 m², pozemek 331 m²	Sopotnice	1 890 000 Kč	prodej	chalupy	110	331	None	Sopotnice	None	1890000.0
3	Prodej rodinného domu 110 m², pozemek 331 m²	Sopotnice	1 890 000 Kč	prodej	rodinného domu	110	331	None	Sopotnice	None	1890000.0
4	Prodej rodinného domu 250 m², pozemek 800 m²	Jičín	5 890 000 Kč	prodej	rodinného domu	250	800	None	Jičín	None	5890000.0
...	...	...	...	...	...	...	...	...	...	...	...
2195	Prodej bytu 3+kk 86 m²	Srní	9 890 000 Kč	prodej	bytu	86	None	3+kk	Srní	None	9890000.0
2196	Prodej bytu 4+kk 125 m²	Karlovarská, Unhošť	7 999 000 Kč	prodej	bytu	125	None	4+kk	Unhošť	None	7999000.0
2197	Prodej bytu 1+1 38 m²	Košická, Praha - Vršovice	5 999 999 Kč	prodej	bytu	38	None	1+1	Praha	Vršovice	5999999.0
2198	Prodej bytu 3+1 81 m²	Rostovská, Praha - Vršovice	14 857 931 Kč	prodej	bytu	81	None	3+1	Praha	Vršovice	14857931.0
2199	Prodej bytu 1+kk 24 m²	Aloise Rašína, Olomouc - Řepčín	3 250 000 Kč	prodej	bytu	24	None	1+kk	Olomouc	Řepčín	3250000.0

2200 rows × 11 columns

3. SQL database

My next step was to create an sql database only from the new columns.

```
import pandas as pd
from sqlalchemy import create_engine

df = pd.read_csv('real_estate_clean.csv')

# Selecting columns to include (exclude the first three)
columns_to_include = df.columns[3:]
df_to_sql = df[columns_to_include]

engine = create_engine('sqlite:///real_estate.db')
```

```
# Writing the DataFrame to the SQL database
df_to_sql.to_sql('czech_RE', engine, if_exists='replace', index=False)

connection = engine.raw_connection()

read_df = pd.read_sql('SELECT * FROM czech_RE', connection)

connection.close()

print(read_df)
```

SQL queries

1. **For each city, calculate the average price_KC and rank the properties within each city based on their price_KC in descending order. Display the property type, city, price_KC, average city price, and the rank.**

```
WITH city_avg_prices as(
      SELECT city, disposition, avg(price_KC) as average_price
      FROM czech_RE
      WHERE RE = 'bytu' and type = 'prodej'
      GROUP BY city, disposition
)
SELECT
      *,
      RANK() OVER(PARTITION BY city ORDER BY average_price DESC) as
rank_price
FROM city_avg_prices
```

11	[illegible]	[illegible]	[illegible]	[illegible]
12	Brandýs nad Labem	3+kk	7383981.5	2
13	Brandýs nad Labem	2+kk	5168462.66666667	3
14	Brandýs nad Labem	4+kk	2481776.0	4
15	Brno	4+kk	15019600.0	1
16	Brno	3+kk	10736394.2	2
17	Brno	3+1	8409800.0	3
18	Brno	2+kk	7643668.14285714	4
19	Brno	*NULL*	5900000.0	5
20	Brno	1+1	5790000.0	6
21	Brno	1+kk	5237982.5	7
22	Břeclav	3+1	4350000.0	1
23	Břeclav	1+1	2969000.0	2

2. Calculate the average price for houses in Praha, Brno and other cities.

```
SELECT
	CASE
		WHEN city = 'Praha' THEN 'Praha'
		WHEN city = 'Brno' THEN 'Brno'
		ELSE 'other_cities'
	END AS 'city_groups',
	round(avg(price_KC),1) as house_selling_price_avg
FROM czech_RE
WHERE RE = 'rodinného domu' AND type = 'prodej'
GROUP BY
	CASE
		WHEN city = 'Praha' THEN 'Praha'
		WHEN city = 'Brno' THEN 'Brno'
		ELSE 'other_cities'
	END
ORDER BY house_selling_price_avg DESC
```

	city_groups	house_selling_price_avg
1	Praha	25712447.1
2	Brno	17789916.7
3	other_cities	9473109.9

3. **Compare average prices for apartments in Brno and Prague according to their layout and calculate percentage difference**

```
WITH brno_prices AS (
      SELECT city, type, disposition, round(avg(price_KC),1) as BrnoAvg
      FROM czech_RE
      WHERE city = 'Brno' and RE = 'bytu'
      GROUP BY city, type, disposition
),
praha_prices AS (
      SELECT city, type, disposition, round(avg(price_KC),1) as PrahaAvg
      FROM czech_RE
      WHERE city = 'Praha' AND RE = 'bytu'
      GROUP BY city, type, disposition
)
SELECT
      b.city,
      b.type,
      b.disposition,
      BrnoAvg,
      round((100 / (BrnoAvg/PrahaAvg)),1) - 100 as PercentDiff,
      PrahaAvg,
      p.disposition,
      p.type,
      p.city
FROM brno_prices b
JOIN praha_prices p ON b.disposition = p.disposition and b.type = p.type
```

	city	type	disposition	BrnoAvg	PercentDiff	PrahaAvg	disposition	type	city
1	Brno	prodej	1+1	5790000.0	4.7	6059249.9	1+1	prodej	Praha
2	Brno	prodej	1+kk	5237982.5	17.0	6130922.3	1+kk	prodej	Praha
3	Brno	prodej	2+kk	7643668.1	12.8	8621919.0	2+kk	prodej	Praha
4	Brno	prodej	3+1	8409800.0	31.2	11036584.2	3+1	prodej	Praha
5	Brno	prodej	3+kk	10736394.2	36.7	14681521.5	3+kk	prodej	Praha
6	Brno	prodej	4+kk	15019600.0	27.3	19120902.8	4+kk	prodej	Praha
7	Brno	pronájem	1+1	13225.0	33.9	17705.0	1+1	pronájem	Praha
8	Brno	pronájem	1+kk	12940.2	32.7	17173.3	1+kk	pronájem	Praha
9	Brno	pronájem	2+1	17775.0	32.5	23552.5	2+1	pronájem	Praha
10	Brno	pronájem	2+kk	19300.0	29.9	25068.3	2+kk	pronájem	Praha
11	Brno	pronájem	3+kk	25300.0	38.7	35084.5	3+kk	pronájem	Praha
12	Brno	pronájem	4+1	25500.0	183.4	72269.6	4+1	pronájem	Praha

4. For each city calculate average price and display only rents of apartments

```
WITH avg_city_prices AS (
      SELECT type, RE, city, price_KC,
            avg(price_KC) OVER(PARTITION BY city) AS city_price
      FROM czech_RE
)
SELECT * FROM avg_city_prices
WHERE RE = 'bytu' AND type = 'pronájem'
```

	type	RE	city	price_KC	city_price
1	pronájem	bytu	Adamov	15800.0	1007600.0
2	pronájem	bytu	Adamov	10600.0	1007600.0
3	pronájem	bytu	Babylon	15000.0	15000.0
4	pronájem	bytu	Benešov	12990.0	7720747.5
5	pronájem	bytu	Beroun	20990.0	7307824.28571429
6	pronájem	bytu	Bor	15000.0	15000.0
7	pronájem	bytu	Brno	14000.0	7065185.51162791
8	pronájem	bytu	Brno	13000.0	7065185.51162791
9	pronájem	bytu	Brno	12400.0	7065185.51162791
10	pronájem	bytu	Brno	13000.0	7065185.51162791
11	pronájem	bytu	Brno	21500.0	7065185.51162791
12	pronájem	bytu	Brno	13400.0	7065185.51162791
13	pronájem	bytu	Brno	19900.0	7065185.51162791
14	pronájem	bytu	Brno	22000.0	7065185.51162791
15	pronájem	bytu	Brno	11000.0	7065185.51162791
16	pronájem	bytu	Brno	12000.0	7065185.51162791
17	pronájem	bytu	Brno	14000.0	7065185.51162791

5. **For Brno and Prague and its city parts show an average price for apartment (for rent or for sale)**

```
SELECT city, city_part, RE, type, avg(price_KC) as average_price,
count(price_KC) as number_of_RE
FROM czech_RE
WHERE (city = 'Brno' OR city = 'Praha') and RE = 'bytu'
GROUP BY city, city_part, RE, type
ORDER BY type
```

	city	city_part	RE	type	average_price	number_of_RE
1	Brno	*NULL*	bytu	prodej	5788000.0	1
2	Brno	Bystrc	bytu	prodej	12990000.0	1
3	Brno	Horní Heršpice	bytu	prodej	5460000.0	1
4	Brno	Husovice	bytu	prodej	10880000.0	1
5	Brno	Kníničky	bytu	prodej	10406512.5	4
6	Brno	Královo Pole	bytu	prodej	6338571.42857143	7
7	Brno	Lesná	bytu	prodej	7600000.0	1
8	Brno	Nový Lískovec	bytu	prodej	6893333.33333333	3
9	Brno	Pisárky	bytu	prodej	15634373.0	2
10	Brno	Slatina	bytu	prodej	*NULL*	0
11	Brno	Staré Brno	bytu	prodej	20000000.0	1
12	Brno	Trnitá	bytu	prodej	9990000.0	1
13	Brno	Veveří	bytu	prodej	8744250.0	4

6. **Find the average RE_area_m2 for each property type and then list all properties that have an RE_area_m2 greater than the average for their type.**

```
WITH avg_RE_area_REtypes AS (
      SELECT type, RE, city, area_m2,
      avg(area_m2) OVER (PARTITION BY RE) AS avg_area_RE
      FROM czech_RE
)
SELECT * FROM avg_RE_area_REtypes
WHERE area_m2 > avg_area_RE
```

	type	RE	city	area_m2	avg_area_RE
1	pronájem	bytu	Praha	86.0	69.7351916376307
2	prodej	bytu	Praha	77.0	69.7351916376307
3	prodej	bytu	Praha	76.0	69.7351916376307
4	prodej	bytu	Praha	79.0	69.7351916376307
5	pronájem	bytu	Ostrava	75.0	69.7351916376307
6	prodej	bytu	České Budějovice	74.0	69.7351916376307
7	prodej	bytu	Praha	87.0	69.7351916376307
8	prodej	bytu	Praha	97.0	69.7351916376307
9	prodej	bytu	Praha	123.0	69.7351916376307
10	prodej	bytu	Praha	91.0	69.7351916376307
11	prodej	bytu	Brandýs nad Labem	98.0	69.7351916376307
12	prodej	bytu	Hodějice	99.0	69.7351916376307
13	prodej	bytu	Brno	95.0	69.7351916376307

7. **Find the cities where the average price_KC is greater than the overall average price_KC for all properties.**

```
SELECT city, avg(price_KC) as city_average, (SELECT avg(price_KC) FROM
czech_RE) as overall_average
FROM czech_RE
GROUP BY city
HAVING avg(price_KC) > (SELECT avg(price_KC) FROM czech_RE)
```

	city	city_average	overall_average
1	Albrechtice v Jizerských horách	9497261.0	6356440.48176583
2	Bedřichov	29766666.6666667	6356440.48176583
3	Benešov	7720747.5	6356440.48176583
4	Benátky nad Jizerou	10756666.6666667	6356440.48176583
5	Beroun	7307824.28571429	6356440.48176583
6	Blatce	6500000.0	6356440.48176583
7	Bohumín	6490000.0	6356440.48176583
8	Bohutín	15499000.0	6356440.48176583
9	Borotín	9750000.0	6356440.48176583

8. **Find the cities where the average price_KC for selling apartments is greater than the overall average price_KC for all selling apartments.**

```
SELECT
	city,
	RE,
	type,
	avg(price_KC) as city_average,
	(SELECT avg(price_KC) FROM czech_RE WHERE type = 'prodej' and RE =
'bytu') as overall_bytu_avg
FROM czech_RE
WHERE type = 'prodej' and RE = 'bytu'
GROUP BY city, RE, type
HAVING avg(price_KC) > (SELECT avg(price_KC) FROM czech_RE WHERE type =
'prodej' and RE = 'bytu')
```

	city	RE	type	city_average	overall_bytu_avg
1	Albrechtice v Jizerských horách	bytu	prodej	9497261.0	8989340.34578147
2	Bedřichov	bytu	prodej	27200000.0	8989340.34578147
3	Chýně	bytu	prodej	9326700.0	8989340.34578147
4	Doksy	bytu	prodej	9763333.33333333	8989340.34578147
5	Horoměřice	bytu	prodej	9410000.0	8989340.34578147
6	Jesenice	bytu	prodej	13200000.0	8989340.34578147
7	Jirny	bytu	prodej	10395000.0	8989340.34578147
8	Kostelec nad Černými lesy	bytu	prodej	11990000.0	8989340.34578147

9. **Which Brno city parts have higher average apt. rents than average apt. Rent in Brno?**

```
SELECT
	city_part,
	avg(price_KC) as city_part_avg,
	round((SELECT avg(price_KC) FROM czech_RE WHERE type = 'pronájem' and
RE = 'bytu' and city = 'Brno'),2) as brno_avg_rent
FROM czech_RE
```

```
WHERE type = 'pronájem' and RE = 'bytu' and city = 'Brno'
GROUP BY city_part
HAVING avg(price_KC) > (
      SELECT avg(price_KC) FROM czech_RE WHERE type = 'pronájem' and RE =
'bytu' and city = 'Brno'
      )
```

	city_part	city_part_avg	brno_avg_rent
1	Bohunice	17000.0	16125.43
2	Chrlice	20950.0	16125.43
3	Ponava	18500.0	16125.43
4	Stránice	34000.0	16125.43
5	Štýřice	21500.0	16125.43