PROJECT DOCUMENTATION

Real Estate Market Analysis – Warsaw and surrounding counties
by Marcin Czerkas

DATE: 26.02.2025

SKILLS & TOOLS COVERED: web scraping, data cleaning, Power Query M, VBA, Excel, Power BI, DAX, data visualization

Dear Reader, thank you for checking out my project!

In this document I am going to introduce you to the project by providing you with a general description as well as all details and technicalities.

I divided this text into **four parts**:

1. **PROJECT OVERVIEW**
   a general introduction to the project and key methodologies I used
2. **DATA**
   web scraping and data cleaning
3. **VISUALIZATION AND ANALYSIS OF RESULTS**
   key insights emerging from the data
4. **GROWTH AND IMPROVEMENT**
   thoughts on what could be improved in the future and ideas on a possible business use of the project

Without further ado, let's dive into the first section.

1. Project Overview

Miejscowość	Dzielnica/Powiat	Cena	Metraż	Cena/m²	Pokoje	Piętro	Biuro nieruchomości	Źródło	Treść ogłoszenia
Bielawa	piaseczyński	2 500 zł	60 m²	42 zł	2			Adres...	
Błonie	warszawski zacho...	2 700 zł	50 m²	54 zł	2			Rentola	Mieszkanie 50,50 m² ...
Błonie	warszawski zacho...	3 000 zł	51 m²	59 zł	2	1	tak	Otodom	Dwupokojowe miesz...
Błonie	warszawski zacho...	3 300 zł	50 m²	66 zł	2	1		Otodom	Błonie mieszkanie be...
Błonie	warszawski zacho...	3 700 zł	48 m²	77 zł	2			Adres...	

The idea of this project was to collect a large enough dataset containing real estate sell and rental offers from the web in order for it to be a representative sample for the analysis of the prices and geographical distribution of the houses for sale and apartments for rental.

To achieve this **I collected the data from 4 different sources** using a quite straightforward web scraping technique. As a result, I was able to produce a dataset of roughly **8000** offers of **houses for sale** and **14000** offers of **apartments for rental** in Warsaw and the surrounding counties. Then, I **cleaned the data** and built an **interactive dashboard**. In effect, the end user has the possibility to deep dive into the data, explore it and draw insights.

I gathered all insights in a short and concise presentation available both in PDF and PPT formats in Polish:

The main challenges I faced was the accessibility of the data, poor data quality and… my geographical knowledge about the districts of Warsaw (which fortunately improved during the project development).

I decided to work on this project for two reasons. Firstly, because I thought the tool I would build might be useful for me privately. Secondly, I wanted to use this opportunity to showcase my skills in data analysis. I thought a combination of something down-to-earth with my passion for technology could make a good project.

2. Data

In this project I did not analyze any ready-to-use dataset from the internet. Instead, I decided to collect the data on my own dataset.

2.1. Web scraping

To access and store the data I used… Power Query. No Python, no APIs, just a simple piece of M code. I wanted to use the simplest possible way (and, simultaneously, prove that it is possible to use Excel for web scraping).

However, there were **three main challenges** that made me get stuck for a while:

Firstly, since the web scraping in Power Query imitates the behaviour of a browser, I was only able to access the first page of each of the search results. So, for example, even if there were 200 results for the search result „Warszawa Bemowo" the website displayed maximally 70 results at one page. I solved it by creating a loop in the query. I used the CSS selectors on the website to identify the place where it says „200 results found". Based on this, I could divide it by the number of items on one page. The result was the number of repetitions for my loop. Each time the link changed by changing the page number.

Secondly, I needed to create another loop to change also the location for which I wanted the results to be displayed. I scrutinized how the links of these websites behave and based on this built another loop, this time in VBA.

Thirdly, the websites can only accept a limited requests in a given period. If my query was sending too many requests, I got blocked and identified as a bot. My solution was as follows: I wrote a VBA macro to refresh only one query at a time. In the query itself I limited the number of pages to be loaded for one location (this way forced me to ignore around 20% of the search results but it did not have a greater impact on my sample).

Below you can find the main parts of my code (both VBA and M):

```
Sub RefreshQueries()

Dim i As Integer
Dim LR As Integer
Dim Location As String
Dim FstTime, SndTime, TrdTime

Application.DisplayAlerts = False
Application.ScreenUpdating = False
Application.EnableEvents = False
Application.EnableAnimations = False

LR = Setup_M4.Range("A1", Setup_M4.Range("A1").End(xlDown)).Rows.Count

Setup_M4.Range("A2:A" & LR).Copy Setup_M4.Range("A" & LR + 2)

'Loop
On Error Resume Next
For i = 2 To LR

    FstTime = Timer

    'Refresh the query
    ThisWorkbook.Queries.FastCombine = True
    ThisWorkbook.Connections("Query - Rentola").Refresh

    SndTime = Timer
    Location = Setup_M4.Range("A2")

    'Delete the last used parameter in Setup
    Setup_M4.Range("A2").ListObject.ListRows(1).Delete

    'Save the workbook to be able to use the next query correctly
    ThisWorkbook.Save

    'Wait 1 minute to avoid sending too many requests to the website and getting blocked
    Application.Wait (Now + TimeValue("0:01:00"))
    TrdTime = Timer

    'Debug
    Debug.Print i & " / Refresh Time: " & SndTime - FstTime & " / Wait Time: " & TrdTime - SndTime & " / " & Location

Next i

Application.DisplayAlerts = True
Application.ScreenUpdating = True
Application.EnableEvents = True
Application.EnableAnimations = True

End Sub
```

```
let
    // PART 1 - WEB SCRAPING

    // Step 1: Define the base URL for the website
    Location = Excel.CurrentWorkbook(){[Name="Adresowo_Setup"]}[Content]{0}[Column1],
    BaseUrl = Text.From("https://adresowo.pl/mieszkania-wynajem/" & Location & "/_1"),

    // Step 2: Extract the total number of pages dynamically
    SourcePage = Web.BrowserContents(BaseUrl & "1"),  // Load the first page to extract pagination info
    TotalResultsText = Html.Table(
        SourcePage, {{"Pagination", ".search-pagination__number-of-pages"}}),
    TotalPages = if Table.RowCount(TotalResultsText) = 0 then 1 else Number.FromText(TotalResultsText{0}[Pagination]),  // Convert the text to a number

    // Step 3: Generate a list of URLs for all pages
    PageNumbers = List.Numbers(1, if TotalPages > 20 then 20 else TotalPages),  // Generate a list [1, 2, ..., TotalPages]
    Urls = List.Transform(PageNumbers, each BaseUrl & Text.From(_)),  // Create URLs by appending page numbers to the base URL

    // Step 4: Define a function to fetch data from a single page
    FetchPage = (url as text) =>
        try
            let
                Source = Web.BrowserContents(url),
                Data = Html.Table(Source, {{"Miasto", ".content STRONG"}, {"Biuro nieruchomości", ".result-info__basic\-\-owner"}, {"Cena", ".text-\[\#c1403d\] + *"}, {"Ulica", ".result-info__address"},
                    {"Pokoje", ".result-info__basic:nth-child(1) *"}, {"Metraż", ".result-info__basic:nth-child(2) *"}, {"Czynsz", ".result-info__price\-\-per-sqm *"}}, [RowSelector=".flex.px-1"])
            in
                Data
        otherwise
            null,  // Return null if the page fails to load

    // Step 5: Apply the function to all URLs and fetch data
    AllData = List.Transform(Urls, each FetchPage(_)),  // Fetch data from each page

    // Step 6: Combine data from all pages into a single table
    ValidData = List.RemoveNulls(AllData),  // Remove null results (in case some pages failed)
    CombinedData = Table.Combine(ValidData),

    // PART 2 - DATA CLEANING

    // Define the list of replacement rules
    Replacements = {
        { " Centrum", "", {"Miasto"} },
        { "Bez pośredników", "", {"Biuro nieruchomości"} },
        { "Oferta biura nieruchomości", "true", {"Biuro nieruchomości"} },
        { "w cenie", "0", {"Czynsz"} }
    },

    // Apply replacements iteratively using List.Accumulate
    ReplacedValues = List.Accumulate(
        Replacements,
        CombinedData,
        (table, replacement) =>
            Table.ReplaceValue(
                table,
                replacement{0},
                replacement{1},
                Replacer.ReplaceText,
                replacement{2})),
    AddedDate = Table.AddColumn(ReplacedValues, "Data pobrania", each Date.From(DateTime.LocalNow()), type date),
    AddedLocation = Table.AddColumn(AddedDate, "Lokalizacja-link", each Location, type text),

    // PART 3 - DATA LOAD

    HistoricalData = Table.PromoteHeaders(
        Excel.Workbook(
            File.Contents("C:\Users\Marcin\Desktop\Varia\Praca\Portfolio\Mieszkania\Mieszkania - baza danych.xlsm"),
            null,
            true
        ){[Item="M3_Adresowo",Kind="Sheet"]}[Data],
        [PromoteAllScalars=true]),
    Append = Table.Combine({AddedLocation, HistoricalData}),
    RemovedBlankRows = Table.SelectRows(Append, each not List.IsEmpty(List.RemoveMatchingItems(Record.FieldValues(_), {"", null}))),

    // Error handling in case of wrong data types
    TypePrep = Table.TransformColumns(
        RemovedBlankRows,
        {
            {"Cena", each try Number.From(_) otherwise null},
            {"Pokoje", each try Number.From(_) otherwise null},
            {"Metraż", each try Number.From(_) otherwise null},
            {"Czynsz", each try Number.From(_) otherwise null}}),
    ChangedType = Table.TransformColumnTypes(TypePrep,{{"Miasto", type text}, {"Biuro nieruchomości", type logical}, {"Cena", type number}, {"Ulica", type text}, {"Pokoje", Int64.Type}, {"Metraż", type number},
        {"Czynsz", type number}, {"Data pobrania", type date}, {"Lokalizacja-link", type text}}),
    ReplacedErrors = Table.ReplaceErrorValues(ChangedType, {{"Miasto", null}, {"Biuro nieruchomości", null}, {"Cena", null}, {"Ulica", null}, {"Pokoje", null}, {"Metraż", null}, {"Czynsz", null}, {"Data
        pobrania", null}}),
    RemovedDuplicates = Table.Distinct(ReplacedErrors, {"Miasto", "Biuro nieruchomości", "Cena", "Ulica", "Pokoje", "Metraż", "Czynsz"})
in
    RemovedDuplicates
```

2.2. Data cleaning

The data I was able to scrape was unfortunately of poor quality. Among the offers, there were some that needed to be excluded, like offers to rent a parking place or to sell a part of a castle (yes!). Apart from that, during the explorative analysis of my dataset, I discovered extreme prices and extreme surface. It turned out that some people, while publishing the offers, input the surface of the whole property, not just the house, while others the surface of the building only. In terms of rental price, some proprietaries included only rent, while others rent with additional fees. Finally, the geospatial data was very messy and needed a standardization – especially since I aimed to visualize the data on the map.

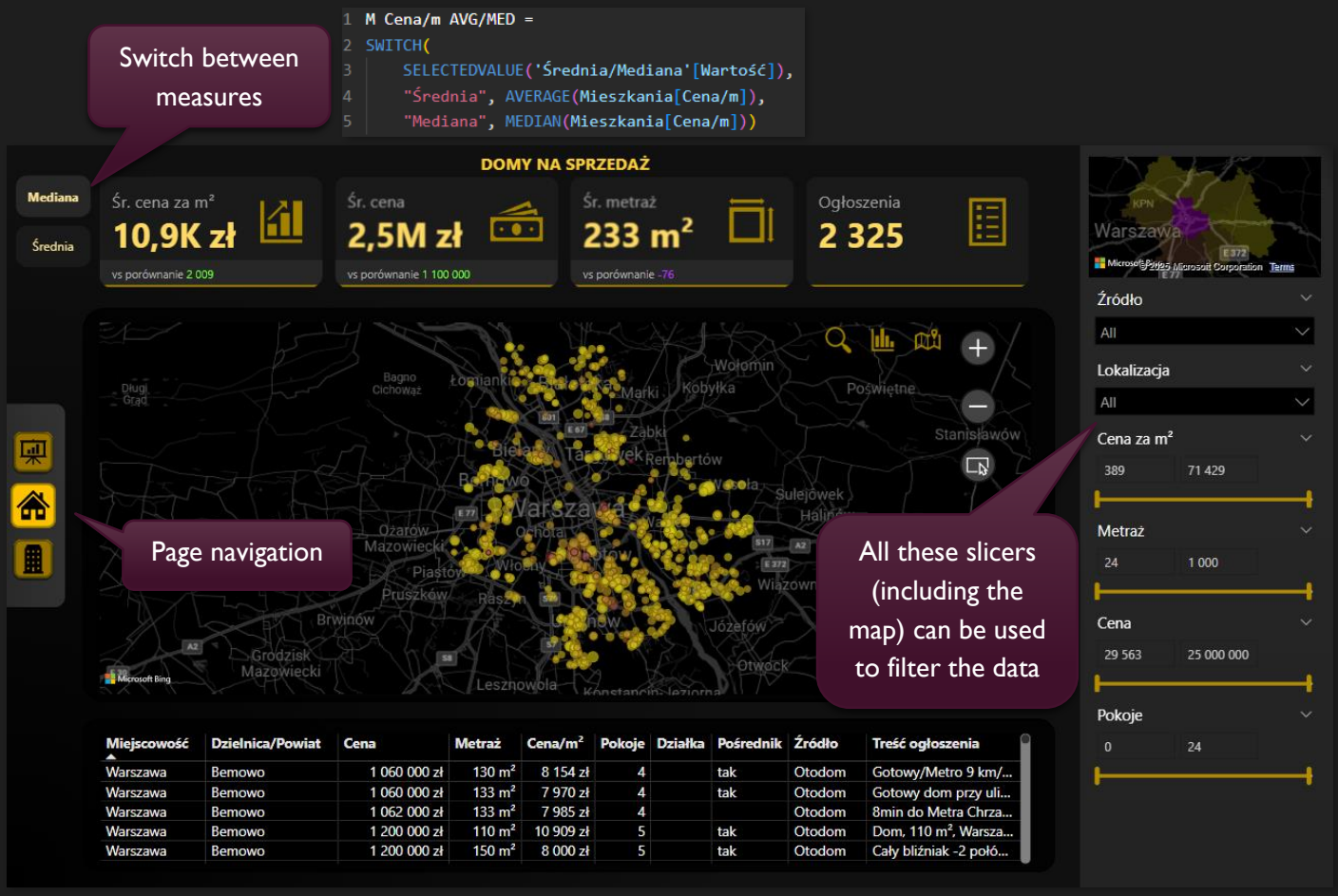
All these and similar data cleaning steps have been done in Power Query in a separate Excel file. If you are interested in checking the details, you can find this file on my GitHub repository dedicated to this project (link below):

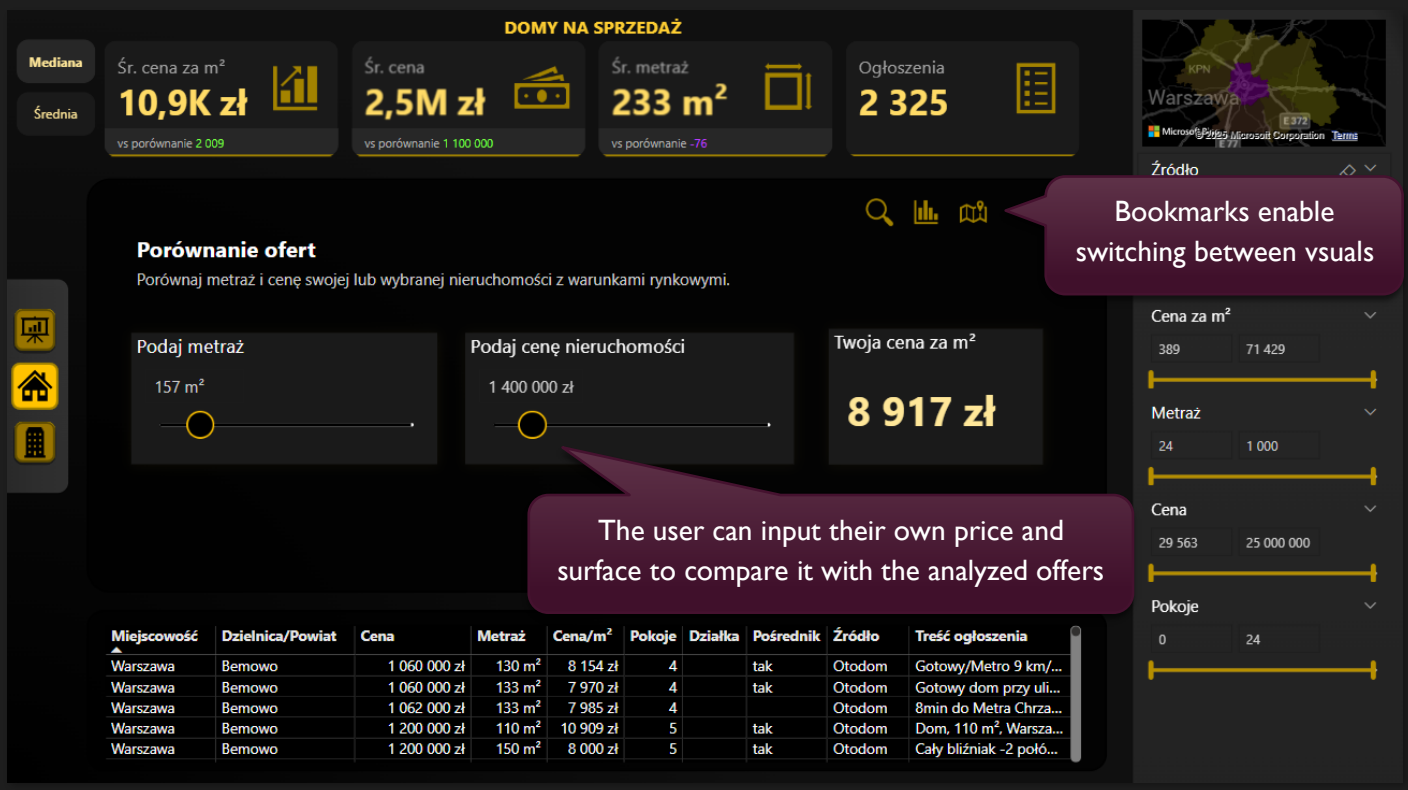
https://github.com/MarcinCzerkas/Real-Estate-Market-Analysis/blob/main/DB_022025.xlsx

3. Visualization and analysis of results

The final part of my project consisted in visualizing the data in Power BI, drawing conclusions and developing an user-friendly tool that could be used by anyone interested in the topic.

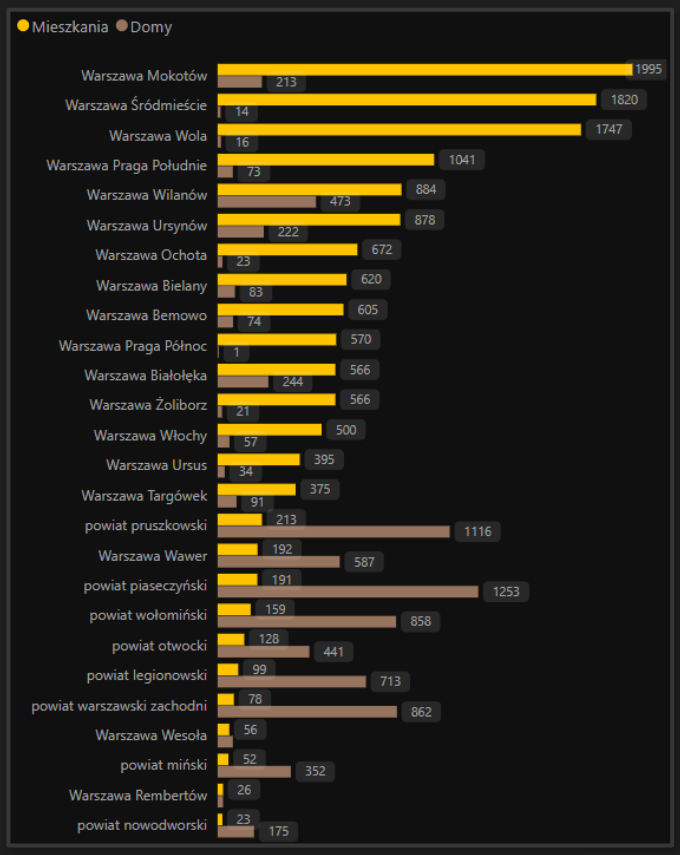
The dashboard itself is quite straightforward. I created a filter pane, included page navigation using buttons and used bookmarks to switch between visuals. Additionally, I added a slicer for the user to decide which measure to apply – average (mean) or median.



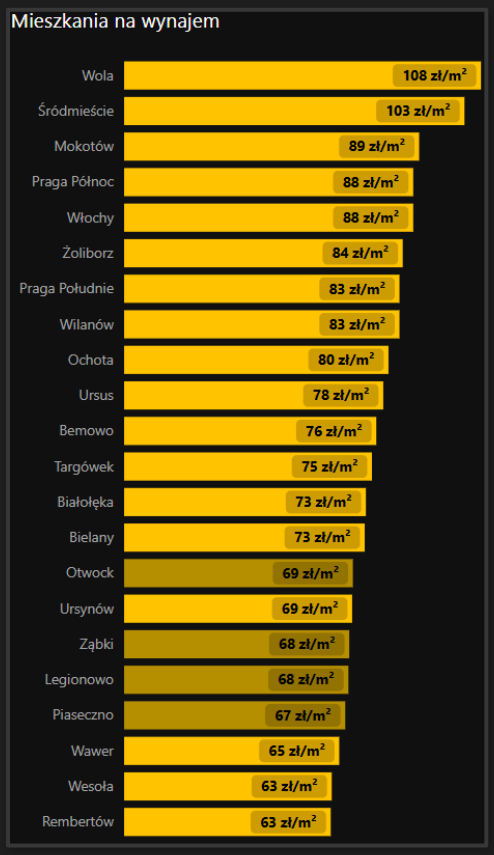


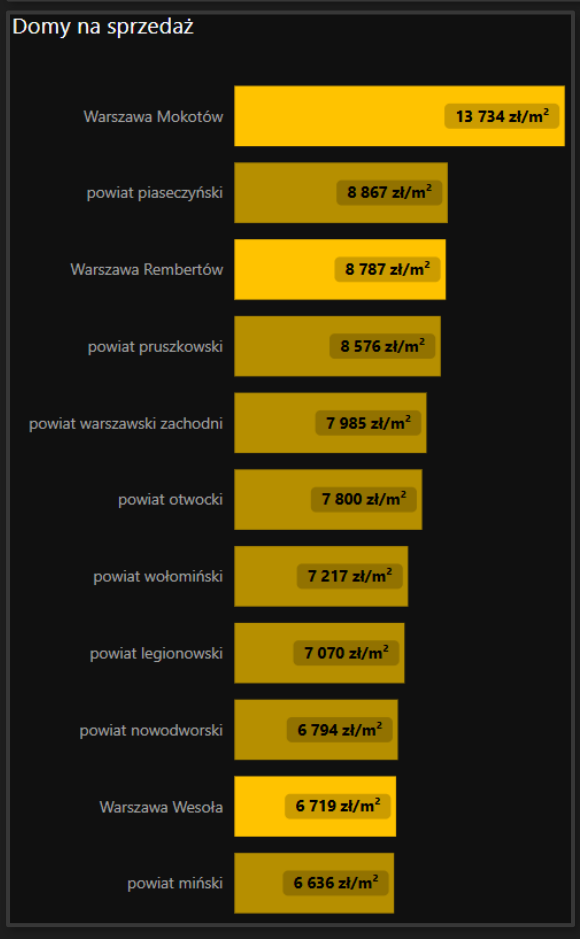
Although the main goal was to give the end user a tool that would enable them to explore the data on their own, I still believe it is a good idea at least to summarize the results in this document. Below you can find the main findings.

1. The most offers of apartments for rent came from the Warsaw districts Mokotów and Śródmieście. The most offers of houses for sale were to find in powiat piaseczyński.



2. In terms of apartments, the highest rental prices per square meter are in the districts of Wola and Śródmieście.





3. The prices per square meter of houses in the suburbs of Warsaw are very close or even lower than in some towns in the surrounding counties.

4. Growth and improvement

What could I improve in my project if I had more time? Which areas would I develop better

if I had enough resources?

One of the things I regret is the fact that Power BI Desktop does not allow me to publish my dashboard to the online service. This is one of the reasons I am considering to recreate the whole tool in Tableau and then publish it to Tableau Public (since it is free and does not require me to have a paid licence). I treat it as an open challenge for the future.

However, the main objection one might have in regards to my project is the fact that I did not use a better web scraping tool. The truth is that at the time I was starting this project my knowledge about Python was still limited. Who knows, maybe in the future I will be able to create a similar analysis, but this time using a much more powerful tool like Python.

Finally, what is the added value of my project?

I think the answer is quite obvious – I developed a tool that might help those who are looking for an apartment to rent or a house to buy. Of course, it is not an exhaustive market analysis. However, the project focuses on the data that such person would like to check anyway – but much slower by checking all websites manually. Thanks to the tool I created all offers are gathered in one place and visualized in a clear way.

It has been my second data project. The first one was a Power BI dashboard build on top of a SQL database created by me to follow the results of a tabletop game. If you are interested, check it out on my GitHub:

https://github.com/MarcinCzerkas/Project-Middle-earth-SBG