Babeş-Bolyai University

The Faculty of Economics and Business Administration

# IT job market in Poland in 2022

Made by

Boróka Nagy

Cluj-Napoca, 2023

# Contents

# 1. Introduction

## 1.1 Presentation of the topic:

The aim of my presentation is to introduce the job market, with a specific focus on the IT sector. As we are in the middle of our final year, many of us have already started working, and those who have not are getting closer and closer to job-hunting. Based on this, I thought it would be interesting to survey which are the most sought-after areas, what knowledge is worth acquiring or developing in order to be able to get a job in our chosen field.

## 1.2 Introduction of the dataset

The dataset I found after a long search specifically focuses on Poland and contains the job advertisements that appeared there. I was particularly happy about this, since I spent 3 weeks there during this summer, and I there is a high likelihood, that I might want live there for a shorter period of time in the future. I created my visualizations from the data of a single dataset, as it analysed the job openings from many different aspects. The dataset consists of 35 columns and 37,788 rows, where each row represents one job advertisement.

(Main dataset: https://www.kaggle.com/datasets/kriegsmaschine/polish-it-job-board-data-from-2022 )

## 1.3 Dataset normalization

Since I found the dataset on Kaggle, it was in a very transparent form at idle, neatly broken down into columns that could be used right away. Therefore the main challenge for me was rather putting the data into the appropriate form, so that I could visualize them properly according to the aspects I envisioned. To work with the data, I used a Google Colab notebook and the Pandas library, which was already presented and used during our classes. In addition, I created the table needed for one of my visualizations in Microsoft Excel.

Among the columns can be found:

- name of the ad (Title)
- name of the city (City)
- country code (Country_code)

• main area of work (Marker_icon)

• type of job - remote/office/partly remote (Workplace_type)

• the experience level – junior/mid/senior (Experience_level)

• date of publication of the job advertisement (Published_at)

In addition to these, several salary-related columns were included, as well as those related to the required prior knowledge. Among these, I tried to use in my visualizations those that could be relevant to me/us regardless of the country we live in, so I did not deal with the salary, because I felt that it is influenced by many factors and depends a lot on the company and the country.

# 2. Visualizations

## 2.1. First visualization:

During the formatting of the dataset, I converted the data of the Published_at column from date-time to date, and then I wanted to replace the country codes with the country names for a more understandable display. To do this, I created a Python dictionary from the two columns of a dataset (Countries ISO Codes) also found on Kaggle, the key values of which were the Alpha-2 country codes, and the values were the country names, then using the .map() function I replaced the country codes with the with country names:

```
[ ]  df4=pd.read_csv("wikipedia-iso-country-codes.csv")
```

```
[ ]  df5.replace(to_replace=df5["Country_code"],
                 value=df4["English short name lower case"])
```

```
[ ]  df4=df4.rename(columns={"Alpha-2 code": "Country_code", "English short name lower case": "Country_name"})
```

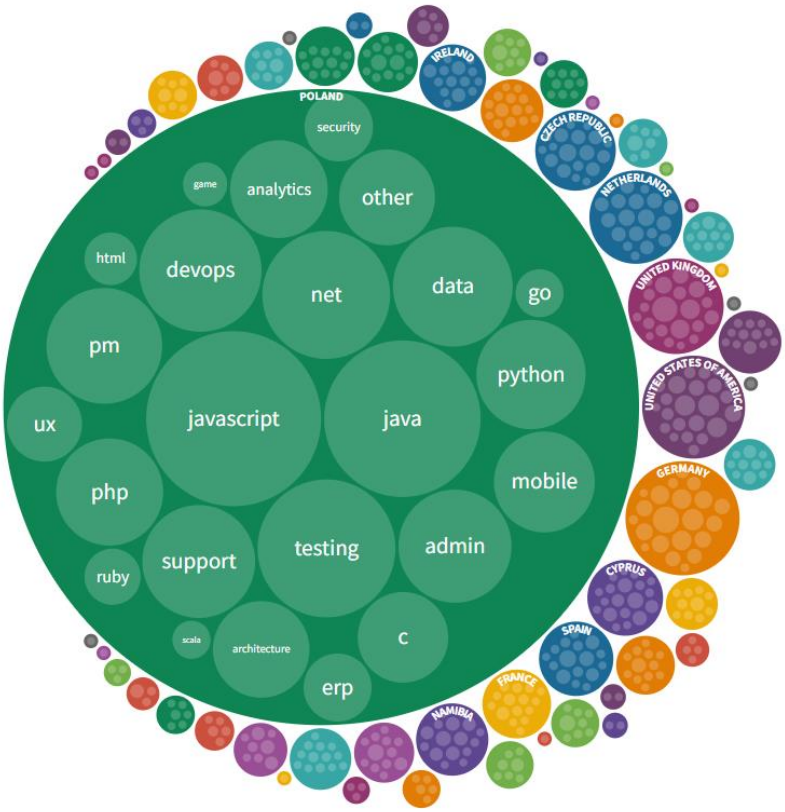
```
[ ]  df4 = df4.drop('Numeric code', axis=1)
     df4 = df4.drop('ISO 3166-2', axis=1)
     df4 = df4.drop('Alpha-3 code', axis=1)
```

```
[ ]  df4_dict=df4.set_index('Country_code')['Country_name'].to_dict()
```

```
[ ]  df5['Country_code']=df5['Country_code'].map(df4_dict)
```

(Countries ISO Codes dataset: https://www.kaggle.com/datasets/juanumusic/countries-iso-codes )

The column I ended up creating was used in the making of a Packed circles hierarchy diagram, where I visualized the main categories of each job advertisement broken down by country, and these were broken down according to the type of job (e.g.: in Poland there were a total of 1,676 Python-centered job advertisements, out of which 1,196 were completely remote, 430 were partly remote, and 50 were office type). The same statement can be viewed for all countries by clicking on the circle marking them.




**Advantage of the visualization**: with the help of a filter, we can also see how the distributions change according to job types (remote, partially remote, office).

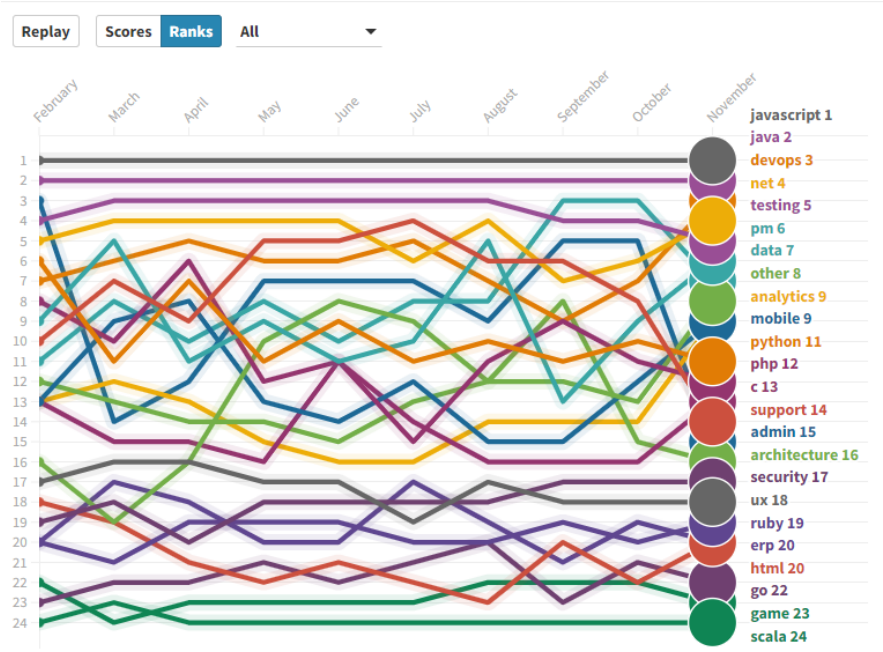
**Disadvantage of the visualization:** not all countries' names are displayed, as most of the ads offer jobs within Poland, but I thought it would be important to present the offers of other countries in at least one of my visualizations, even if the figures are much lower than for Poland.

## 2.2. Second visualization:

My second diagram is a Line chart race diagram, in which I examined how the job advertisements of each area changed from February to November, and which area became the most popular. For this I had to extract the month from the Published_at date. Since these appeared as numbers, I again needed a dictionary where the keys were the month numbers and the values were the month names, and again I used .map().

```
[ ] months_dict={1:'January',2:'February', 3:'March',4:'April',5:'May',6:'June',7:'July',8:'August',9:'September',10:'October',11:'November',12:'December'}

[ ] df13['Month_name'] =df13['Month']

[ ] df13['Month_name']=df13['Month'].map(months_dict)
```

There is also a filter on the visualization, with the help of which we can look at the sequence in terms of the number of job advertisements or just the position of the areas.




**Advantages of the visualization:** It shows well the change of positions during the year.

**Disadvantages of the visualization**: It is a bit difficult to follow the 24 areas at the same time, but with the help of a filter you can see the change of only one area at a time.

**Alternative representation**: perhaps a Bar chart race, but only 10 categories out of 24 can be displayed at the same time there.

## 2.3. Third visualization:

My third visualization is also a hierarchy type Radial tree. Here again, I present the number of job advertisements broken down by month according to experience level (junior, mid, senior). Here, too, there is a filter where you can select a specific month and when it comes into focus, the data is divided according to another dimension: number of permanent jobs and fixed-term jobs. Thus, the final result, summarized in one sentence, would look roughly like this: In May, the number of total junior positions were 722, out of which 444 were permanent positions and 278 were for a fixed period.

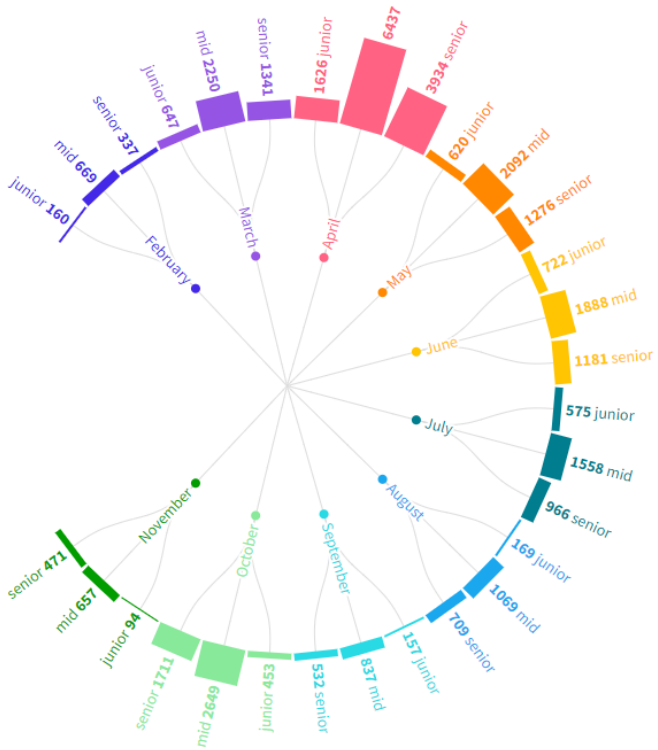
I displayed the months in the same way as in the previous visualization, and then adjusted the data using the .groupby() method:

```
[ ] df3=df2.groupby(["Month","Month_name","Experience_level", "if_permanent"])["if_permanent"].count()

[ ] df3.head(3)

Month  Month_name  Experience_level  if_permanent
2      February    junior            Not permanent     64
                                     Permanent         96
                   mid               Not permanent    258
Name: if_permanent, dtype: int64

[ ] df3.to_csv("PermenentJobsPerMonth.csv",index=True)
```

**Advantages of the visualization:** you can clearly see what the essence of the visualization is

**Disadvantages of the visualization:** the distribution according to Permanent/Not permanent categories is only displayed if you click on a specific month

**Alternative representation**: Any other hierarchy type diagram




## 2.4. Fourth visualization:

My fourth visualization is a word cloud, which is used quite often, but since I wanted to show the frequency of occurrence of words, it seemed the most appropriate. The basis of my visualization was the skills considered more important by the job advertisements, which I grouped with the help of .groupby() and set in descending order according to the number of occurrences. I put the most common 180 words in the word cloud.

```
[ ]  df7=df7.groupby(["skills_name_0"])["skills_name_0"].count().reset_index(name="count")
```

```
[ ]  df7=df7.sort_values(by=['count'],ascending=False)
```

```
[ ]  df7=df7.head(180)
```

```
[ ]  df7.to_csv("SkillsWords_1.csv",index=False)
```


## 2.5. Fifth visualization:

My fifth and final visualization is a Sunburst hierarchy diagram, which again focuses on the most important or most valued skills. The visualization shows the 3 most frequently mentioned skills of all 24 main areas, and by clicking on an area, you can see how many times each skill was mentioned.

Perhaps it was the most difficult to get the data into the right form here, as the data had to be grouped several times, and the values had to be set in descending order according to occurrence, and then the first three from each category had to be cut.

```
[ ]  df6=df5.groupby('Marker_icon', group_keys=False).apply(lambda x: x.sort_values('count', ascending=False)).groupby('Marker_icon').head(3)
```

**Advantages of visualization:** All data can be seen broken down into categories.

**Disadvantages of visualization**: The dimensions of the articles do not show anything.

**Alternative representation: Any** other hierarchy type diagram.


## 3. Conclusions

In conclusion, I can say that I really enjoyed the process of the visualization and I became aware of the fact, that I managed to choose a very good dataset, several times, as both the proximity of the topic and the accessibility of the data helped me a lot in my work. I also found the results interesting and I think that although these visualizations do not represent the reality with 100% accuracy, they can be good guidelines for anyone interested in the topic.

The link to my Flourish visualizations: https://public.flourish.studio/story/1873320/