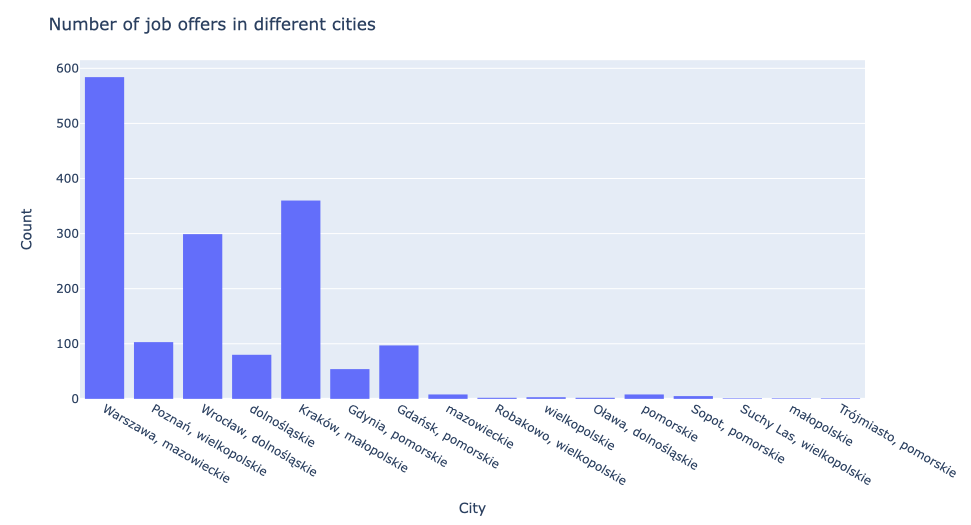
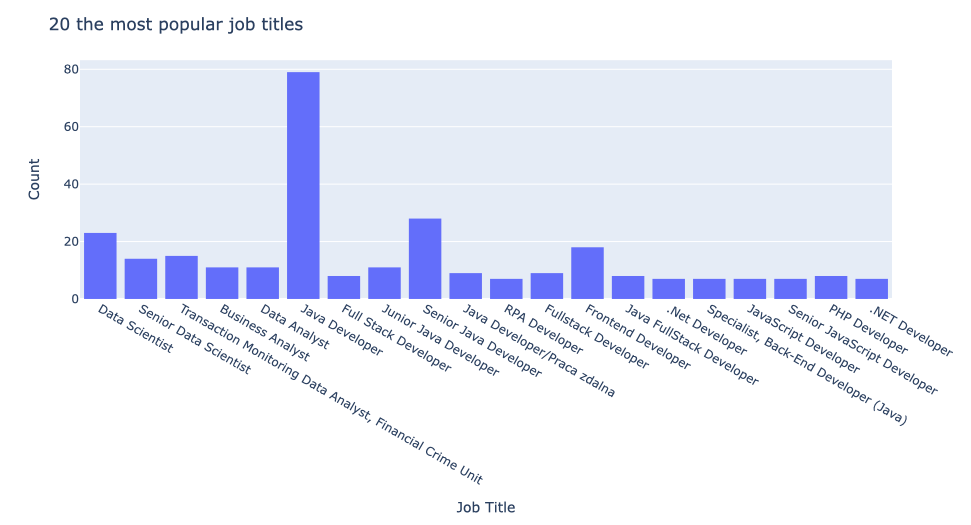
**Web Scraping Project - Spring 2021**

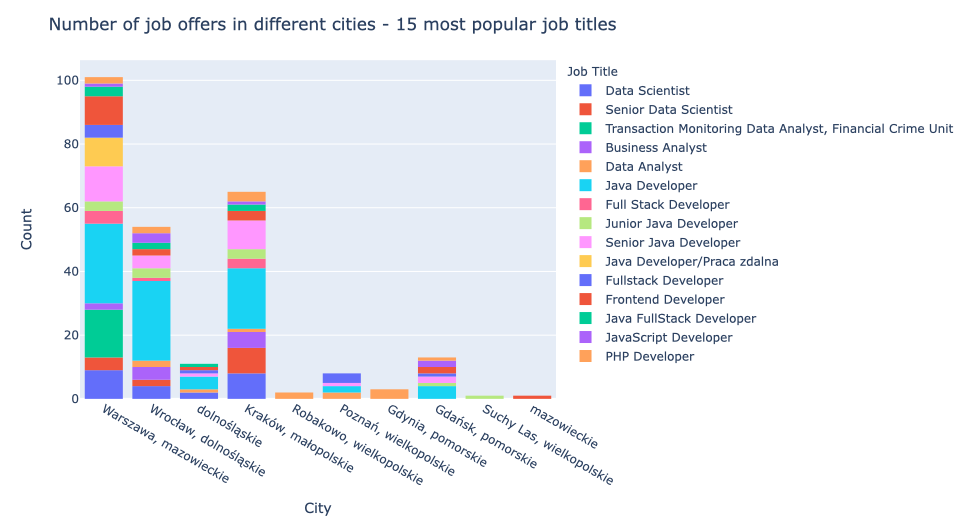
1. **Names and ID’s:**
   1. Ewelina Plachimowicz (397115),
   2. Raffaele di Costanzo (434917).
2. **Description of the topic and the web page -** we decided to scrap **indeed.com** web page. This page contains job offers, opinions about employers and the salary comparison tool. We decided to use only one part - job offers search. The user of a web page can look for a job after some filters - job title (e.g. data scientist), city (e.g. Warsaw, Masovian district), type of contract (full-time, internship, part-time, contract or temporary) and how old the job post is (last 24 hours, last 3 days, last 7 days or last 14 days). Our main goal is to scrap job offers with their titles, companies names, cities, summary descriptions and dates of publication - with filters defined by user and csv or xlsx output with all data.
3. **Short description of scraper mechanics -** during that project we used all three scraping tools - Beautiful Soup, Scrapy and Selenium. Firstly, the user defines filters to job searching - job title, city, type of contract and ‘age’ of post with the job offer. The user needs to provide at least one argument, if not our code returns error. After declaration of inputs, our code searches indeed.com with given filters and scrap job titles, companies names, cities, summary descriptions and dates of publication for all offers posted on the web page. After scraping, code saves all data in a csv or xlsx file on the user's computer.
4. **Technical description of the output -** our output is in csv or xlsx format and given observation is one job offer. We have six columns:
   1. job title - e.g. Data Scientist,
   2. company name - e.g. Goldman Sachs,
   3. city - e.g. Warsaw,
   4. summary description - e.g. ‘Experience in working with data and delivering business-changing analytics, data science and/or marketing personalization (consumer- centric data)’,
   5. ‘age’ of the given offer - e.g. today, 3 days ago, 30+ days ago,
   6. link - link to the offer.
5. **Data Analysis:**

Our output can be used for further analysis of the labor market. If someone wants to analyse IT market in main cities in Poland, he can use filters to define typical job titles for IT e.g. ‘Machine Learning Developer’, ‘Data Scientist’ , ‘Data Analyst’, ‘Java Developer’, ‘JavaScript Developer’, cities - Warsaw, Poznań, Wrocław, Kraków, Gdańsk, type of contract - e.g. full-time and ‘age’ of offer - eg. last 14 days (to be as up-to-date as possible) and scrape data for these filters. After it, he can merge it by rows and analyse, e.g. apply basic summary statistics (how often some company appears, how many offers for a given job title, how many offers for a given city etc.) or visualise it. Below some example of our analysis for these filters.

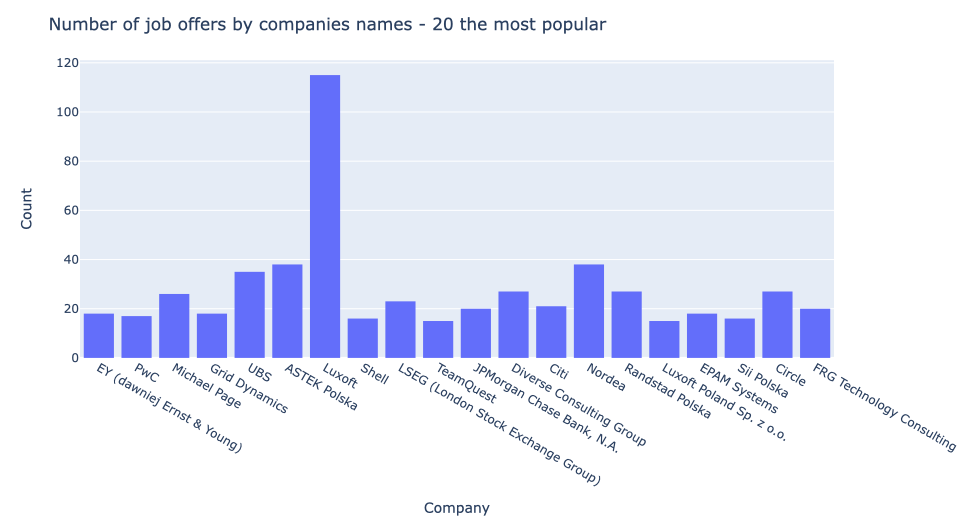


The best place to find a job in IT is Warsaw. Moreover, we have quite many offers in Wrocław and Kraków. Gdańsk and Poznań are less comfortable for programmers.





Generally employers look for Java developers (or any other developers) or Data Scientists.



Luxoft has the biggest number of offers, next one is ASTEK, Nordea and UBS.

1. **Segregation of duties:**
   1. Ewelina Plachimowicz - Scrapy part, documentation part - instructions, explanation, data analysis,
   2. Raffaele di Costanzo - Selenium and Beautiful Soup part.