Visualizing the Job Market

using online postings

The JobInsider Group

Cesar Ernesto Illanes Argote,
Yuanlong Li, Tianzong Zhang

COM-480 Data Visualization

2025 Spring
IC-EPFL

Part 1. Introduction

Job Market. What can we learn from the job listings?

1. Overview.

The job market, and the global economy as a whole, has gone through their fair shares of ups and downs. **How is the job market perspective for a job seeker currently?** This is a question that is especially vital to us young graduates.

The motivation behind this data visualization project is to provide insights into the evolving job market, helping individuals and organizations understand which job titles, locations, and industries are experiencing the highest demand, as well as how compensation and benefits are distributed across different sectors. The analysis will also explore the prevalence of remote work and how companies differ in their internship offerings and employee benefits.

2. Methodology.

a. Data Acquisition.

To implement our idea, we began by searching for publicly available datasets containing job listings. We came up with the LinkedIn Job Postings (2023 - 2024) Dataset on Kaggle, which contains 124,000+ job postings listed in 2023 and 2024 in the United States. While ideally we would like to have a dataset that focuses on Europe, we did not succeed in finding such. According to the documentation, the dataset was obtained by scraping all the job postings on LinkedIn by using a job scraper. It contains multiple features regarding every LinkedIn job listing, including salary, work position, company name, job title, job description, and industry.

To parse the geoinformation data associated in the dataset, we additionally use the USA state geojson data, which contains the shapefile, population and area of every state in the US.

b. Data Exploration.

To make use of the data, we first perform data exploration to get familiar with the data. We inspected different features of the dataset:

- **Salary**.

  Around 70% of the job listings do not specify the salary. In fact, in the US, disclosing the salary range is only voluntary instead of obligatory except for in a couple of states. Nevertheless, we can still count the job listings for every state, and we still obtain enough listings with salary to perform the statistics on salary.

  For those job listings with a specified salary, we investigate the distribution of the normalized salary column. It turns out that the normalized salary has a very large variance and a significant amount of outliers. This may be due to the algorithm for calculating the normalized salary. As a result, we use median instead of average in the data processing part below.

- **Listing dates**.

  Contrary to what was documented in the dataset page, the listing dates only span from 12.2023 to 04.2024 instead of the whole years of 2023 and 2024. Moreover, the listing dates are not uniform across the said timespan, and most of the job listings date between 03.2024 to 04.2024. This hints that the dataset is not comprehensive enough, but we perform the statistics between the timespan regardless.

- **Other Features**.

  We also investigated other columns such as the job title, industry and work type. We refer to the data exploration codes for further detail.

c. Data Processing.

It is clearly impossible and counterintuitive to visualize all the 124k job postings one by one on a demo. As a result, we need to perform aggregation statistics for the job postings. We use pandas for data processing. Specifically, as discussed in Section 2.b, we count the number of job postings, as well as the median salary for each state and each date. For those job listings without salary or other features, we fill in -1 as default values instead of removing them, so that statistics for other features can still be exploited.

The processed data is stored as a JSON file, which is to be consumed by d3.js in the frontend. You can find the data here.

Part 2. Visualization

Job Market. How do we present the job listings?

3. Designing the Visualization.

a. General Considerations.

Our goal was to build a clear, interactive dashboard to help job seekers explore trends in job demand and salaries across the U.S. We started with a simple wireframe to define the core components: a time filter, job title search, a U.S. map, and trend charts.

We used choropleth maps to show geographic differences, gradually improving them by adding interactivity (tooltip/popups) and filters by industry or data type. A timeline slider and line+bar charts allowed users to track job demand and salaries over time.

To present more detailed insights, we created a carousel layout showing distributions by industry, company, and top-paying sectors. We also prioritized clean visuals, consistent color schemes, and easy navigation for a user-friendly experience.

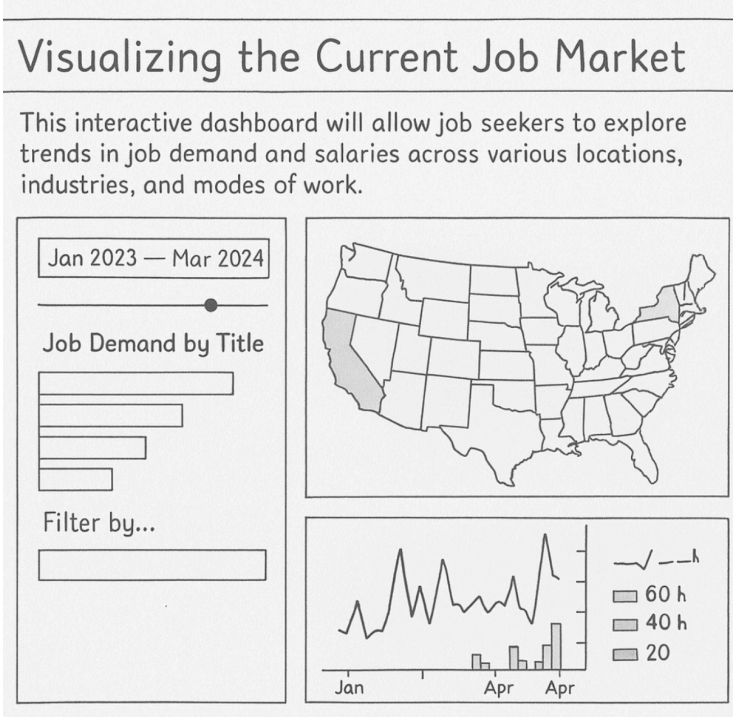


Fig. 3.a. scratch

b. Redesigning the Map.

The map is used to show the statistics of the job listings for each state in the US given a user-selected date range. In our map, we choose to visualize the median salary within the

date range, and the map is shown in the format of a choropleth map. In the draft version, we have set up a minimal example of the map (see Fig. 3.b.1). In our opinion, the Openstreetmap basemap color scheme distracts users from the visualization, so we opt for BlackWhiteMap by StadiaMaps to improve the design. In addition, we changed the layout of the map layers so that the name of the states are clearly visible. The improved map is shown in Fig. 3.b.2. However, we recognized that the black and white color scheme, while effective for reducing visual distraction, is not widely adopted and may feel less visually appealing to users. To address this concern, we maintained the clarity benefits of the monochrome base layer while also overlaying a standard map layer option. This hybrid approach resulted in our final map design that successfully balances both visual clarity and aesthetic appeal.

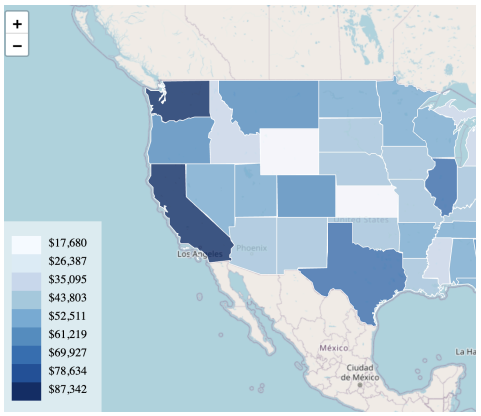
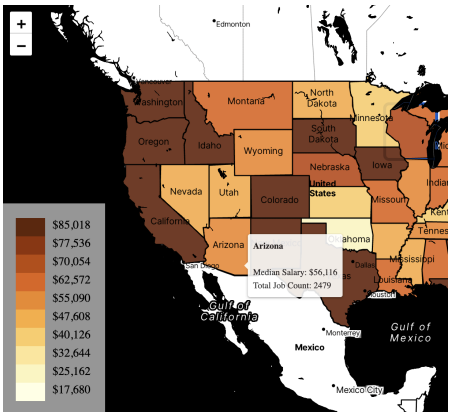


Fig. 3.b.1. Map as in Draft.



→ **Fig. 3.b.2.** Improved Map

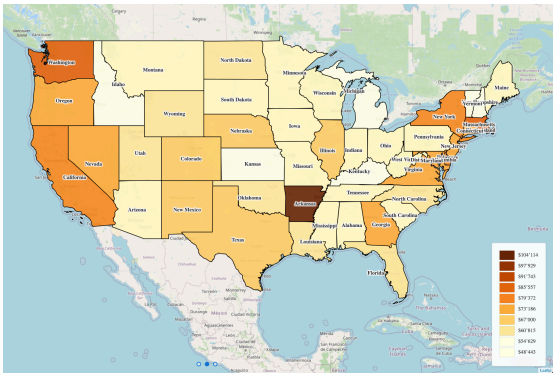


Fig. 3.b.3. Final Map

c. Redesigning the Chart.

The chart is used for showing the time series of the statistics (job count, median salary) per day given the user-selected date range and state. The median salary is visualized as a line chart and the job counting is visualized as a bar chart. Fig. 3.c.1. shows the chart as in the prototype. We improved the visualization by adjusting the bar width and interpolating

the line chart. To further smoothen the chart, we discarded the data where there are less than 3 job postings, as the median salary is prone to be affected by outliers under such circumstances.

Upon further reflection, we realized that these temporal trends provided limited practical value for job applicants, as users could hardly extract actionable insights from them. Additionally, combining the choropleth map with a timeline slider resulted in fragmented data presentation that hindered users' ability to understand the situation. Therefore, in our final version, we decided to separate these two analytical approaches. The choropleth map now focuses exclusively on the geographic distribution of job opportunities, enhanced with the ability to filter by industry sectors. To capture temporal trends in the job market, we developed a dedicated view that aggregates how different industries and companies vary in their salary offerings and hiring volumes over time. To make the entire interface more appealing and informative, we also extensively incorporated various UI widgets, such as tooltips, to provide users with additional contextual information. The final result of this redesigned approach is illustrated in Figure 3.c.3.

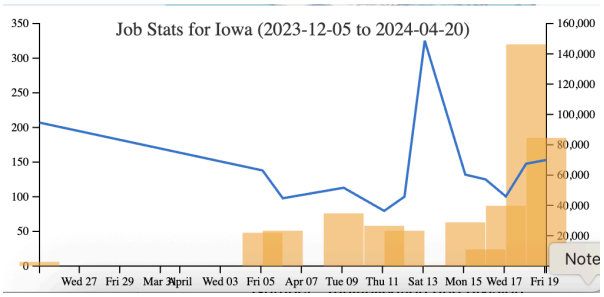
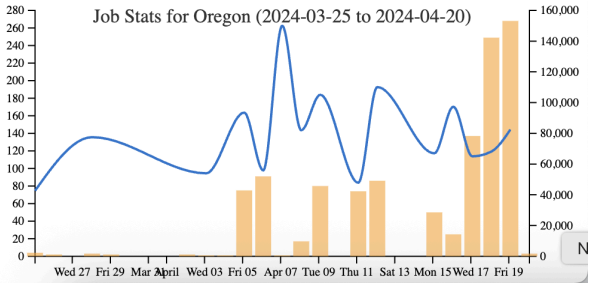


Fig. 3.c.1. Chart as in Draft.



→ **Fig. 3.c.2.** Improved Chart.

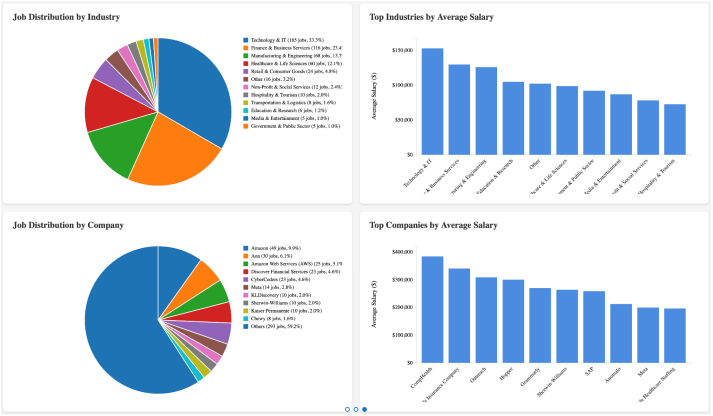


Fig. 3.c.3. Final Chart.

4. Implementing the Visualization.

We use python/pandas to extract, clean and preprocess the data.

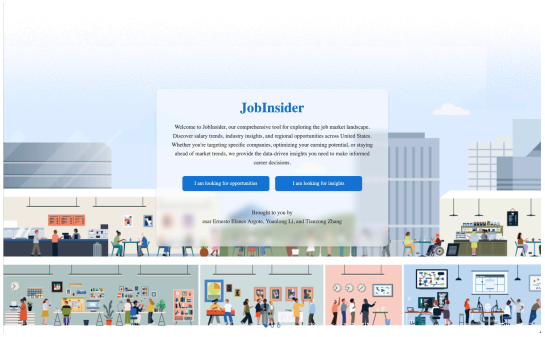
We use html/css/javascript to implement the frontend webpage.

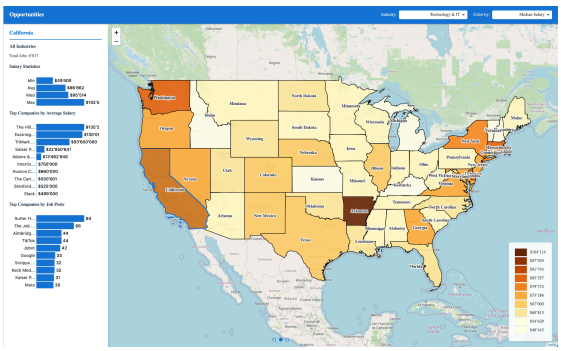
We use d3.js to implement data visualization on the frontend.

We use leaflet.js to implement map visualization.

5. Results.

We ultimately developed a three-page design structure. The first page serves as an entry point and introduction to the entire application, while the remaining two pages offer distinct analytical perspectives. The second page is designed specifically for job applicants, providing comprehensive geographical and industry-sector statistics of the job market. This view is particularly valuable as it highlights the most influential companies that offer either the highest salaries or the largest number of job openings within each sector. The third page presents a more analytics-oriented view, enabling users to examine job market behavior across various time periods and grasp higher-level insights into how different industries and businesses develop over time.





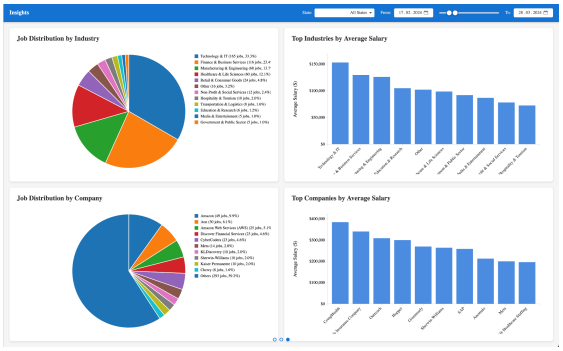


Fig. 3.c.3. Final website.

Part 3. Assessment.

6. Peer assessment.

As a team: We searched and decided which dataset we would use.

Cesar: Wrote the dataset description and related work in Milestone 1, Milestone 2 report and part of the Process Book. Also wrote the first draft of the background information on the page. Also worked on another implementation that is on “another_design” branch on GitHub.

Yuanlong: Performed dataset exploration in Milestone 1. Enhanced the webpage design in Milestone 2. Mainly designed the final website, made screencast, and wrote part of the Process Book in Milestone 3.

Tianzong: Searched for dataset and wrote report for Milestone 1. Cleaned dataset, drafted the initial visual design, and implemented the prototype of the website in Milestone 2. Optimizing maps/charts, and mainly drafted the Process Book in Milestone 3.