Data Science Job Market Analysis

Bridget Crampton, Ian Shigley, and Micah Cheng

The explosion of computer advancement and data collection in the 1980s led to a growing need for labor in various professions such as Data Engineers, statisticians, Data Scientists, etc. The importance of these positions continues to grow as we are just hitting a new wave of transformative learning, generative AI. Taking Moore's law into account, we can assume that the market for these jobs will increase exponentially as this technology advances. To test this we are putting This paper examines multiple aspects of the data science job market through a data set of five thousand data science jobs collected from online job forums from 2020 to 2022 with their respective locations, salaries, etc.

1. INTRODUCTION

Since the emergence of computer science and programming in the 1960's, the market for jobs in adjacent fields have exploded. One of these subsets in the market is the field of data science which helps reinforce the natural human inclination to take what has happened and use it to predict what will happen. With the advent of the internet, data scientists were able to collect an exponentially larger amount of data with which they could predict trends and behavior for businesses. This has allowed for many innovations, particularly in the way businesses develop the marketing, segmenting, and selling of a product. With the recent pandemic and new generative AI affecting the stability of certain job markets, we hope that the analysis of the shifts in certain aspects of data science jobs will help determine where the field will go in the future which will help us tremendously as future members of the workforce.

2. METHODS

We used the libraries Matplotlib, pandas, and Seaborn mainly to assist with concise coding with less code used. While we learned Matplotlib and pandas in class, we learned Seaborn through the website "GeeksforGeeks" to make graphs easier to code since we had a large dataset ("Python Seaborn Tutorial"). We used pandas (pd) to read the file for our separate data analyses.

Salary by Job Category Over Time

The Pandas .groupby() function was a key tool in visualizing salary against variables like time, company location, job category, and job title. For example, we created a line plot to visualize how salaries in USD changed over time across different job categories. To prepare the data, we used the `.groupby()` function along with `.mean()`. First, we grouped the data by work year and calculated the average salary, storing the result in `salary_trend`. Then, we grouped the data by both work year and job category to calculate the average salary for each combination, storing it in `salary_trend_by_job`.

To plot the data, we used Matplotlib. After initializing the graph with `plt.figure()`, we used `plt.plot()` to add the main trend line. Then, we plotted the salary trends for each job category on the same graph using a `for` loop. This approach allowed us to overlay all job category trends on a single graph.

Salary Box Plots

To analyze salary trends based on experience levels, we filtered the original DataFrame (`df`) to create separate DataFrames for Entry-level (EN), Senior-level (SE), and Mid-level (MI) employees using the `.isin()` function. For the entry level, we selected rows where the experience level is 'EN' and saved the result in `df_EN`, similarly for Senior-level ('SE') and Mid-level ('MI'), saving them as `df_SE` and `df_MI`, respectively. These filtered DataFrames allow for focused analysis of salary trends by experience level.

For salary distribution by job category and location, we created functions to generate box plots. The `plot_salary_distribution_by_l`

`ocation` function visualizes salary distributions by company location, taking the dataset, plot title, and filename as inputs. It uses Seaborn's `boxplot()` to display salaries (`salary_in_usd`) grouped by location (`company_location`) with a 'whitegrid' style, adds labels and a title, and saves the plot as a PNG file. This function was applied to `df_EN` and `df_SE`, generating separate plots titled "Salaries by Company Location (EN)" and "Salaries by Company Location (SE)."

Similarly, the `plot_salary_distribution_by_job_category` function uses Seaborn and Matplotlib to create box plots visualizing salary distributions across job categories. It sets a 'whitegrid' style, creates a box plot with salary (`salary_in_usd`) on the x-axis and job category (`job_category`) on the y-axis, and colors the boxes by category using the hue parameter. We applied labels, a title, and custom font sizes and the plot is saved as a PNG file. This function was also applied to `df_EN` and `df_SE`, creating and saving separate plots for each dataset with appropriate titles and filenames.

Lastly, we created a boxplot to show the distribution of salaries by year. This code creates a box plot to visualize salary distribution by work year. First, `sns.set(style='whitegrid')` sets the background style to a clean grid. Then, `plt.figure(figsize=(8, 5))` initializes the plot with a specified size. The `sns.boxplot()` function plots salary (`salary_in_usd`) on the y-axis against work year (`work_year`) on the x-axis, using the `df` DataFrame for the data. The plot is customized with a title, x-axis label, and y-axis label, all with adjusted font sizes. Finally, the plot is saved as a PNG file with `plt.savefig()` and displayed with `plt.show()`.

Salary Bar Graphs

For the bar graphs, we first calculated the average salary by job title for the entry-level dataset (`df_EN`), sorted the results in descending order, and created a bar plot using Seaborn to visualize the salary distribution by job title. The plot was saved as an image file. For the wage premium calculations, we computed the average salary by job title for three datasets: entry-level (`df_EN`), mid-level (`df_MI`),

and senior-level (`df_SE`). We then merged these datasets and calculated the wage premium for the mid-level dataset by dividing its salary by the entry-level salary. A bar graph was created to show the wage premium by job title and saved as an image. Similarly, we calculated the median salary by job category for each dataset and followed the same process to calculate and visualize the wage premium for mid-level relative to entry-level by job category. Both bar graphs were saved as images for further analysis.

Pie Charts

One of our objectives was to use pie charts as a way to see if there was an increased amount of jobs in a specific title, or job category. If there was an increased amount, this would mean there would be more demand or popularity for this job and that students should seek out these skills. In order to do this, we used .value_counts() to count the frequency of each job category/job title in the column compared to all the others ex: AI or Data Analyst, and converted it into a pie chart by using .plot(kind='pie') and adjusted the chart using autopct(the splitting of the chart), fig size, color, etc. We showed its output using plt.show and labeled it using plt.title, plt.ylabel, and plt.xlabel. The pie chart is saved as a PNG file with `plt.savefig()` and displayed with `plt.show()`.

User Input Graphs

Another objective was to create a code that would produce a graph from the dataset, based on what the user of the code chose, first the x variable, y variable, then the type of graph(line plot, boxplot, barplot, or scatter plot). After a lot of trial and error in adjusting fig size, we decided on fig size (14, 6) that suits most of the graph types depending on the # of variables. We plotted the figure using plt.figure and further gave the plot detail using the if, elif, and else statements to create different outcomes depending on the user input. There was also an error message that would show up if the user input was incorrect. Seaborn was included to code the graphs with less code and create a white grid for coherent backgrounds. The def print_instructions(): function was created to print instructions for the user to follow. Then, the parameters of the function are determined by x_column=input(), graph_type=input(), and the user's input. Finally, the function is activated by create_custom_graph(df, x_column, y_column, graph_type).

3. RESULTS AND DISCUSSION

The results of this study revealed several conclusions concerning salary in the data science work field.

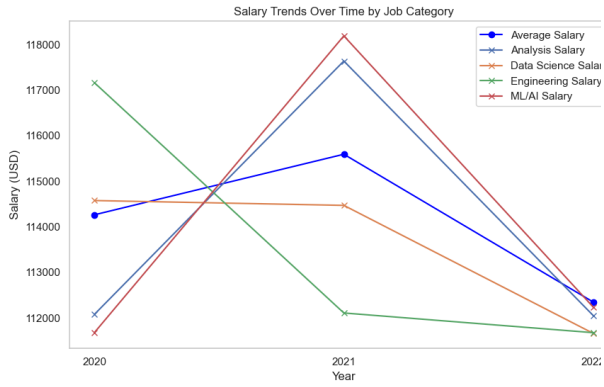

Figure 1. Salary Trends Over Time

Figure 1 shows how the salary for data science rose from 2020 to 2021 in all job categories except Engineering, which has been decreasing since 2020. From 2021 to 2022 all 4 job categories decreased in salary, on average from about $115500 to $112500, a difference of $3000. One conclusion we can draw from this is that salary fluctuates over the years but the average difference is only about $3000. We can also deduce that Engineering salary has gone down more steadily on average than the other job categories, however, we cannot make any concrete conclusions seeing as the time frame is so short.

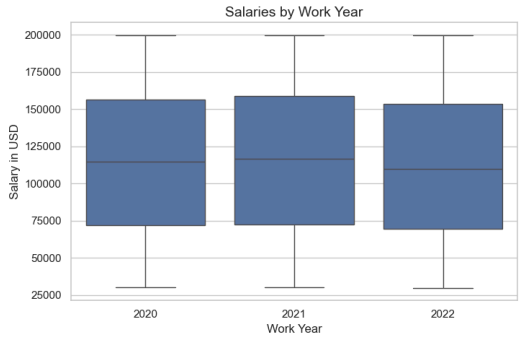

Figure 2. Salaries by Work Year

Figure 2 also supports the conclusion that salary fluctuates but only in very small differences as the upper bound and median in the box plots change very minutely.

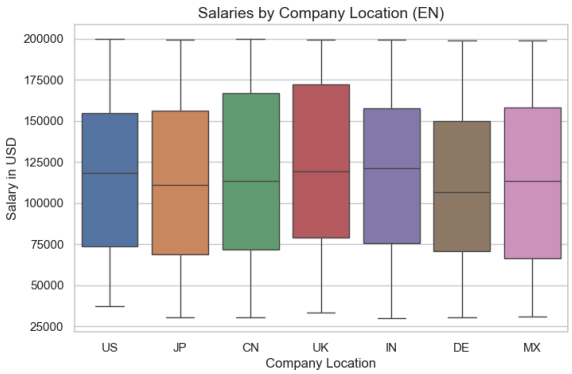

Figure 3. Salaries by Company Location (EN)

Figure 4. Salaries by Company Location (SE)

Both Figure 3 and Figure 4 reflect the range of salaries by company location, with Figure 3 showing this for only entry-level positions and Figure 4 showing only senior-level positions. Figure 4 highlights significant variations. Japan (JP) emerges as the top-paying location with the highest median salary, while countries like Mexico (MX) and India (IN) have relatively lower median salaries. Additionally, countries such as Germany (DE) and Japan show a broader salary range, indicating opportunities for higher pay at the senior level. On the other hand, locations like Mexico and India display lower minimum salaries, which may reflect regional salary disparities and cost-of-living differences. This chart underscores the influence of geographic location on SE compensation, with developed economies offering significantly higher pay.

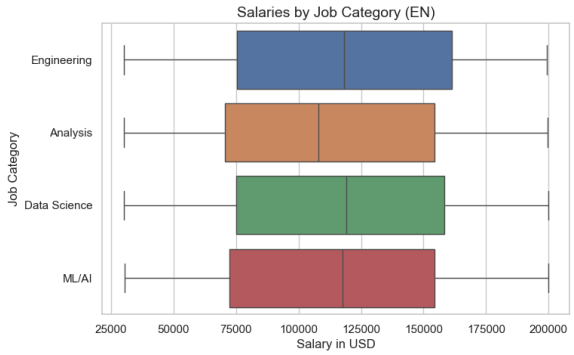

Figure 5. Salaries by Job Category (EN)

Figure 5 focuses on entry-level (EN) salaries across different job categories. Machine Learning/Artificial Intelligence (ML/AI) and Engineering roles lead with the highest median salaries, reflecting the high demand for technical expertise. In contrast, Analysis and Data Science positions have slightly lower median salaries, though they still offer competitive pay. ML/AI roles also exhibit the broadest salary range, suggesting diverse compensation based on specialization and skill level. By contrast, Analysis and Data Science positions display narrower ranges, indicating more consistency in entry-level pay for these roles. Overall, technical fields like ML/AI and Engineering offer more lucrative opportunities at the EN level compared to non-technical or hybrid roles.

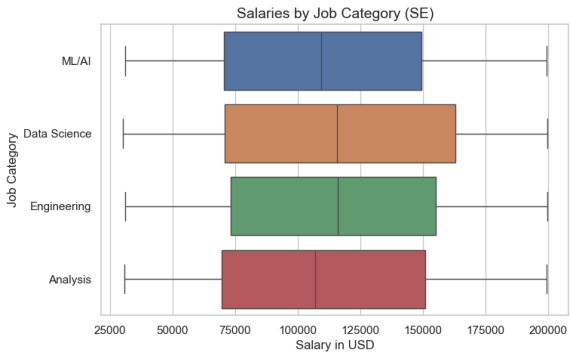

Figure 6. Salaries by Job Category (SE)

Figure 6 follows similar trends as Figure 5 with ML/AI and Data Science offering higher medians and wider ranges, reflecting

a premium on expertise. Analysis roles have the lowest median and tighter ranges.

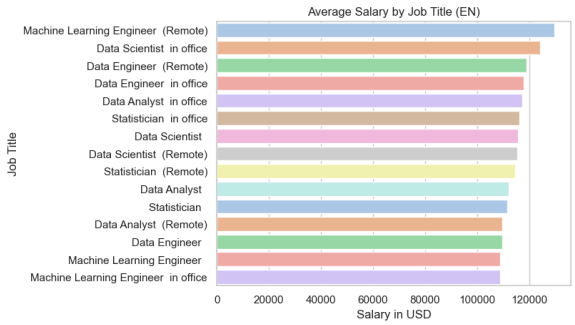

Figure 7. Average Salary by Job Title (EN)

Figure 8. Wage Premium by Job Title

Figure 8 ("Wage Premium (MI) by Job Category") indicates the relative wage premiums associated with various job categories. Analysis roles exhibit the highest wage premium, followed closely by ML/AI and Data Science roles, which have similar levels of premium. Engineering roles, while slightly lower than the other categories, still exhibit significant wage premiums. This suggests that skills related to analysis and ML/AI are currently highly valued in the market.

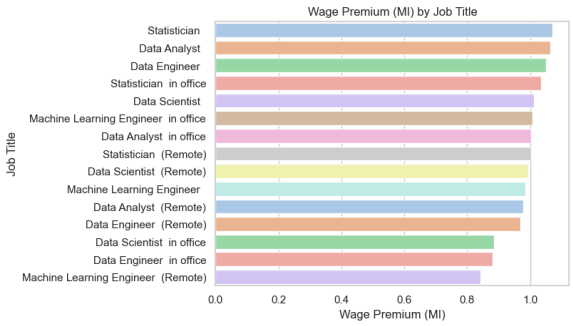

Figure 9. Wage Premium by Job Title

Figure 9 ("Wage Premium (MI) by Job Title") provides a detailed breakdown of wage premiums across specific job titles and working arrangements (e.g., remote vs. in-office). Across titles, remote roles seem to maintain a slightly higher wage premium than their in-office counterparts, highlighting the increased value of remote flexibility in some industries. Machine learning engineer roles (both remote and in-office) tend to have higher premiums, followed by Data Scientist and Data Engineer roles. Statisticians and Data Analysts have comparatively lower premiums, emphasizing that advanced technical and domain-specific expertise tends to command higher wages.

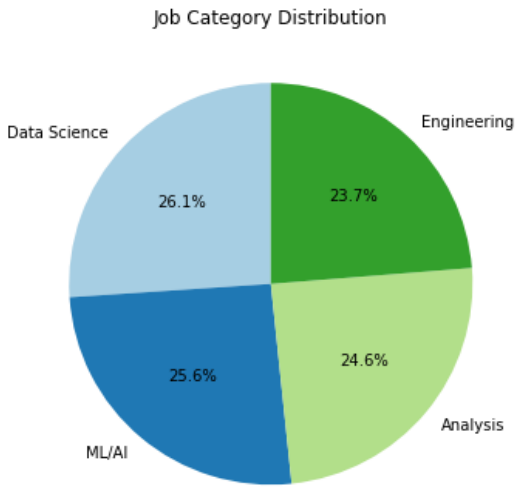

Figure 10. Job Category Distribution

Figure 10 shows the balance of job distribution for each category of data science job. This was intended to see if one particular category was in more demand or was more popular so the users of our code would know to pursue the skills needed to be in that field. Unfortunately, the distribution was very evenly spread with Data Science having the greatest amount of jobs at 25.6% and Engineering having the least amount of jobs at 23.7%.

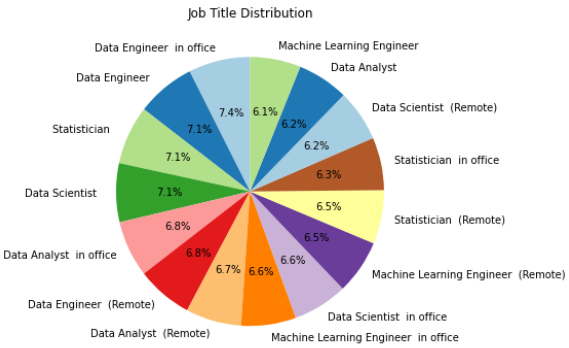

Figure 11. Job Title Distribution

Figure 11 shows the balance of job title distributions in the general Data Science job category. We wanted to display this graph to see if there was a higher demand for any job so users of code could keep in mind to develop certain skills, however all of the jobs were evenly split in amount, with Data Engineer in office at 7.7%, and Machine learning engineer as the lease at 6.1%, so no plausible conclusions could be made from the graph.

4. CONCLUSION

Overall we concluded that although our coding would be effective with a different dataset with more data, the dataset that we used needed to be bigger to make any valid conclusions especially. Our dataset was also hard to work with since there was only 1 reliable numerical variable, salary in USD. Our user-input graph code could be useful in analyzing similar types of datasets with any topic and can be used to analyze any set of numerical and categorical variables. Additionally, the findings underscore the importance of developing specialized technical skills, particularly in ML/AI as they consistently exhibit the highest wage premiums and salary ranges. Remote flexibility appears to be increasingly valued, suggesting a trend toward more equitable

opportunities across locations. Moving forward, expanding datasets and incorporating non-salary variables, such as skill requirements or industry-specific demand, could provide deeper insights into the dynamics of the data science job market.

Work Cited

U.S. Bureau of Labor Statistics. “Data Scientists.” *Occupational Outlook Handbook*, U.S. Department of Labor, 6 Sep. 2023, https://www.bls.gov/ooh/math/data-scientists.htm.

“Data Scientist Job Market: Trends, Statistics, and Future Predictions.” *365 Data Science*, https://365datascience.com/career-advice/data-scientist-job-market/.

“A Brief History of Data Science.” *DATAVERSITY*, 20 Feb. 2020, https://www.dataversity.net/brief-history-data-science.

Brsahan. “Data Science Job Dataset.” *Kaggle*, https://www.kaggle.com/datasets/brsahan/data-science-job.

“Python Seaborn Tutorial.” *GeeksforGeeks*, https://www.geeksforgeeks.org/python-seaborn-tutorial/.