GROUP A: DATA SCIENCE JOBS IN THE US

Group Members:
Silvia Sunseri (ss6105@columbia.edu)
Davide Vaccari (dv2438@columbia.edu)
Dylan Rosenthal (dr3118@columbia.edu)
Bengusu Ozcan (bo2297@columbia.edu)

In this document we are going to explain how we realized every single graph that can be found in our final project. For all the graphs, except for the last two, we used the data about Data Science Job Posting on Glassdoor. It was collected by web-scraping job posts from Glassdoor for data science jobs. For the last two graphs instead, we used the 2018 Kaggle Machine Learning & Data Science Survey, the most comprehensive dataset available on the state of ML and data science. As a disclaimer, the dataset on Kaggle about Glassdoor job postings did not specify the time frame it was collected from. Despite trying to reach out the provider of the data, we could not clarify this. Despite not knowing the exact time frame, the data still provides useful information across the attributes e.g. geography, job title, salary… For the visualization in which we merged this data with Google trend data, we assumed this data as a yearly pull, simply because there is no significant seasonality in data related jobs and the data could be representative of a yearly pull.

First graph “Average Salary by job”

The first graph that we created describes the average salary by job. First of all, we cleaned the data by grouping job titles into the same categories. For example, if a job description contained the word “Analyst”, we decided to assign this particular job to the “Data Analyst” category. If a job description contained the word “Architect”, we decided to assign this particular job to the “Data Architect” category. I then calculated the average salary by job title and created a column graph with job title on the x axis and average salary on the y axis.

Second graph “Data Related Job Posting Salaries per Sector”

This graph was created by grouping the job postings data set per employer sector, and then summarizing the total number of openings as well as the average salary. For the average salary calculation, the column that is already provided as “average salary” per job posting is leveraged. Since there were 22 sectors, the graph is further simplified by filtering the sectors that have at least more than 3 job openings, in order to make the graph more representative for the general sector. The number 3 is chosen by looking at the whole range of job openings, with the objective of including up to 15 sectors in the visualization for simplification purposes. Finally, the graph is transformed into an interactive one.

Third graph “Google Searches vs Job Openings”

This graph was created by merging the Google search trends and number of job openings for certain job titles in our data set. Job titles “Data Scientist”, “Data Analyst”, “Machine

Learning Engineer" and "Data Engineer" are chosen based on their popularity in the job postings and also personal interests of our group members. Google search data is pulled by leveraging Google Trends API, in the format of "total number of searches in the past year for specific keywords" with the breakout of each US state. The intention was to find out the recent popularity of data related jobs across states. Initial job posting data set is grouped by state and job title, limiting to the job titles selected earlier. In order to create separate graphs combined in a faceted view, the original data set is filtered down to 4 sub datasets according to each job title. Then for each job title, Google search and job posting data sets are merged, yielding a data table of the number of job openings and Google search trends across each state.

After being merged, the data is simply summarized for total number of searches and job openings per state. 4 individual graphs are generated and turned into interactive plots. Then these four graphs are bundled together as subplots.

First map for the Number of Openings

We decided to take a look at the job openings geographically. We wanted to see both where each job opening was located and to group them to find where most of them are. The grouping that made the most sense was by State. In fact, someone might be interested in moving to a certain State instead of a specific City. Alternatively, one might want to know how many data science job openings are available in her State as she might want to stay close to home. Therefore, the first map shows both the number of openings grouped by State and the openings for cities.

We used dplyr library to group_by the openings by states and by cities. We used count to count them and arrange to arrange them. Then, we used the DT library to create two datatables. Then, we geocoded the cities location applying the geocode function from ggmap through purrr library function map_df. We used Google's API. The shapefiles for the United States were downloaded from (https://www.census.gov/geographies/mapping-files/time-series/geo/carto-boundary-file.html).
We made the popup text and we merged the dataframes. We created the color bins based on the number of openings for each state and used the "RdBu" palette.
Leaflet library is perfect for interactive maps, so we chose that one. Thanks to "addLayersControl" we could fit two maps in one and make it interactive. This way the viewer can choose whether to look at the openings per state or per city.

Second map for the Average Salary

In the second map we were interested in showing how salaries differ from State to State and which are the cities that pay the most. To make the visualization easier to interpret, we decided to differentiate the cities whether they were above or under the national median salary for the job. Moreover, the size of the circle increases as the average salary per city increases.

The code for this map is very similar to the one above. We grouped the data by states and cities. Here, instead of count, we used summarize as we wanted to compute the average salary as a mean of the average salaries for each opening by states and cities. Then, we rounded the digits to two decimal places. We wanted to add another feature for the cities to make the map more readable and give more information. We used ifelse function and mutate to add a feature whether the city average salary was above or below the National median value. Then we geocoded as above, added the popups, and merged the dataframes. Here two palettes were made, one for the states in green (as the dollar), and one for the cities, green if above the National median, red if below. Here we also used the leaflet library. We wanted to focus more on the cities so we made the city map to be the default one and added a legend. Moreover, the size of the circle is proportional to the city's average salary. This can be done through “radius” in “addCircleMarkers”.

Wordclouds

We used wordclouds as we wanted to analyze the descriptions of the job openings to better understand what skills and abilities are desired by potential employers.
The first thing we did was to clean text strings from job descriptions, removing URLs, digits, punctuation, and white space. Next, we tokenized words from job descriptions into individual rows using “Unnest_Tokens" function. Next, we created a list of keywords (focusing on those that indicated a key skill, tool, or task) to focus on for analysis, removing general words like “science” or words with a specific industry focus. Next, we uploaded an excel file of words of interest and did a left join with the unnested job description data to remove words that were not useful for our analysis. After that, we used the “Group_by” and “Summarize” arguments to create a new data frame counting the number of times that each key word appeared in all data science job descriptions. Next, we built a word cloud of just “skills” showing the most common words included in data science job descriptions. We then created a term document matrix from the unnested job description data counting the number of times that each key word appeared in a job description for data analyst and data scientist roles only. We then created a comparison cloud of most common skill and tool words that appeared in data analyst versus data scientist job descriptions. Lastly, we created a comparison cloud of most common tools that appeared in data analyst versus data scientist job descriptions.

Fourth graph “Average Data Scientists’ Salary by Educational Level and Gender”

This graph was created by using the data from our second dataset. The first thing we did was to keep only the observations from the United States since our focus is this exact country. We also kept only the observations for male and female respondents with a Master’s Degree vs a Doctoral Degree as we wanted to see the difference in salary for these categories.
We then calculated the average salary with the formula (minimum salary + maximum salary)/2. Every observation was an individual respondent of the survey. The graph we wanted to design was a violin graph with average salary by educational level and gender. I found a function that was able to design only half of the violin graph and put the two halves together. This is what the

function that we used does. Finally, we were able to design the violin graph where on the x axis we put the gender of the respondent, on the y axis the average salary, and we filled the violin graph according to the educational level attained.

Fifth graph “Average Data Scientists’ Salary by Age and Gender”

The last graph we designed reports the average data scientists’ salary by age and gender. The only recoding we had to do in this case was to adjust the age categories. In the original dataset, we had too many age categories, and therefore, we decided to group them into bigger categories so that at the end there were only 10 age categories. The bar chart we designed has on the x axis the age groups, on the y axis the average salary, and the bins are colored according to the sex of the respondents.