1. Problem Statement

The goal of this project is to help immigrants select the location, job title, and companies for the most popular jobs in the USA. The most popular jobs are the ones which are high in the demand and pays the most. The demand for each of the job title is determined based on the number of jobs present in the market in the past and the salary paid for that role. Power BI is used to clean, transform and visualize the data to generate insights to answer the above questions.

2. Data Source

Data for project was collected from kaggle (https://www.kaggle.com/datasets/jboysen/us-perm-visas). The data contained information about the visa application and job for which the application was submitted. This data was from 2014 to 2018 only. I went to the original source of the data (https://www.dol.gov/agencies/eta/foreign-labor/performance) to get the data between 2015 to 2002.

3. Data Cleaning and Transformation

The original data had 154 columns and 800 thousands rows, only 20 columns which were of interest to answer the business questions were kept. Data from 2015 to 2022 was imported into 8 different tables in Power BI. The following steps were applied to clean and transform the data:

1. Change the data types of the columns to Date/Number/Text based on the information in the column.

2. Power BI could not detect the type of data for some of the columns because the data was input incorrectly. This issue was removed by only selecting the rows where the number of employees in a company were greater than 50.

3. After changing the data types, the names of the columns were changed to be consistent in all the 8 tables.

4. All 8 tables were appended to a single table.

5. Created a new column called processing time by subtracting application received date from decision date.

6. There was one negative value of the processing time because of the incorrect date formats in the columns. This value was replaced by a positive value after correcting the date format.

7. The null values in the processing time was replaced by the average of the column.

8. The null values in the received date column was replaced by the calculated values (decision date – processing time).

9. The null values in the employer establishment date was replaced by the most common value in the column.

10. The null values in the PW soc code, and title were replaced by the string “NA”.

11. The null values in the Prevalent Wage (PW) Unit of Pay column was replaced by year if the value of PW was null.

12. Replaced the null values in Prevalent Wage (PW) column by the minimum wage 30K.

13. Replaced the null values in the Prevalent Wage Unit of Pay by “hour” if PW value is less than 30, else bi-week if the PW wage is less than 1200 and for other values with year.

14. Removed all the rows with null values where wage offer values were missing. As wage offer would be our target value so we should not introduce any kind of bias here.

15. Followed the similar procedure to replace the null values in the columns wage offer unit of pay and wage offered as followed for the PW wage unit of pay and PW.

16. Replaced the null values in state, city, minimum education required to “NA”

17. Replaced the null values in the minimum required experience to “N”(No).

18 Replaced the null values in the required experience month (number of months) column to zero if the value in minimum required experience is “N” otherwise replaced with the mean of the column.

19. Replaced the null values in country of citizenship, visa type to “NA”

20. Calculated a new column called annual prevailing wage (PW) using the column prevailing wage based on the units of the payment. For example for monthly unit of payment, the annual wage can be calculated by multiplying the PW wage with 12.

21. Similarly calculated the offered annual wage based on the units of the wage pay.

22. Converted all the text columns to upper text.

23. Some of the job titles were misspelled and repeated, changed those to the correct title using the replace query.

24. The work state column had both the full name and abbreviation for the states. I replaced the abbreviations with the full name of the state using if statements.

25. Removed the duplicate rows from the data.

26. Filtered the rows where annual salary is less than 1M and greater than 15K to remove outliers.

27. Created a new column called job group based on the job title column to categorize the jobs into a more general category. For example, I put software developer, software engineer, computer engineer into a same category called software developer.

4. Data Visualization

First, I visualized the number of jobs for each job title and their corresponding income. Software developer and Business analyst jobs were two of the most common job titles. I also visualized the median income for different job titles. Next, I visualized the location for most number of jobs that offers the highest salary.

The Software Developer and Business Analyst roles were most in demand jobs so I choose these two roles for further investigation. I analyzed the variation of median salary over time for these two jobs. I also analyzed the top paying companies for these roles. Lastly, I analyzed the number of jobs for these two roles at different organizations.

5. Automating the Process

To automate the process of importing the files, I developed a function in power query. This function takes the path of the folder, name of the files to be imported and name of the excel sheets to be imported as arguments. This function returns each excel sheet as a table in power bi. We can check the code for the function by going to power bi>view>advanced editor.

To automate the refreshing of data in real time, we can use the data refresh option in Power BI. We can setup a specific frequency that will be used to refresh the data in this option. More on this can be found here(https://learn.microsoft.com/en-us/power-bi/connect-data/refresh-data).

6. Next Steps

I did not get enough time to put in visualization part of the data. In next 2 days, I plan to visualize the data further using different measure created by DAX functions.