***Data Science Project Lifecycle – Individual Coursework***

* Chosen Dataset is Baby Names for Girls in England and Wales: [babygirlsnames](file:///C:\Users\bhavj\OneDrive\Desktop\Bhavjot%20UNI\Second%20Year%20(Level%205)\Data%20Science%20Project%20Lifecycle\Individual%20Coursework\girlsnames2023.xlsx)
* Video Presentation: <https://drive.google.com/file/d/1I7hNn6lg0zIdeUU1CAijVPowxmp2RrRp/view?usp=drive_link>
* Streamlit App: <https://dsplindividualcw-xhsyzcbeohr3qgkdswvtt5.streamlit.app/>
* GitHub Repository: <https://github.com/Bhavjot-L/DSPL_Individual_CW>

***Aims and Development Methodology***

The aim of this project is to explore trends and patterns in baby girl name choice across England & Wales using the official dataset of top 100 names. By analysing how name popularity varies by region and changes over time. An interactive dashboard will be developed using Streamlit to present these insights in a user-friendly format. Also, video demonstration will be provided to clearly to highlight the key findings and the functionality of the dashboard.

*Objectives:*

* To identify the most popular baby girl names in England & Wales for recent years.
* To analyse how the popularity of specific name has changed over time.
* To compare naming trends between regions (e.g., England vs Wales)
* To develop an interactive dashboard that allows users to explore and filter the data
* Create a video walkthrough of the dashboard, demonstrating the key insights.

*Development Methodology:*

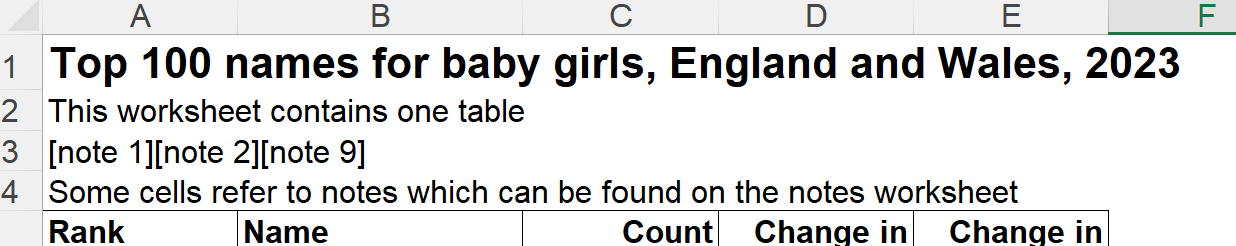
For this project, I utilised the Agile methodology because it supports flexibility, and continuous improvement which is usually ideal for developing an interactive dashboard where feedback and visual changes plays a big role. Instead of going through the project in one go, I changed it to subtasks like data cleaning, exploratory analysis, visualisation design and testing. The purpose of this is to ensure the tasks are easier and make steady progress while regularly testing and evaluating what was working and what needed changes.

***Requirements***

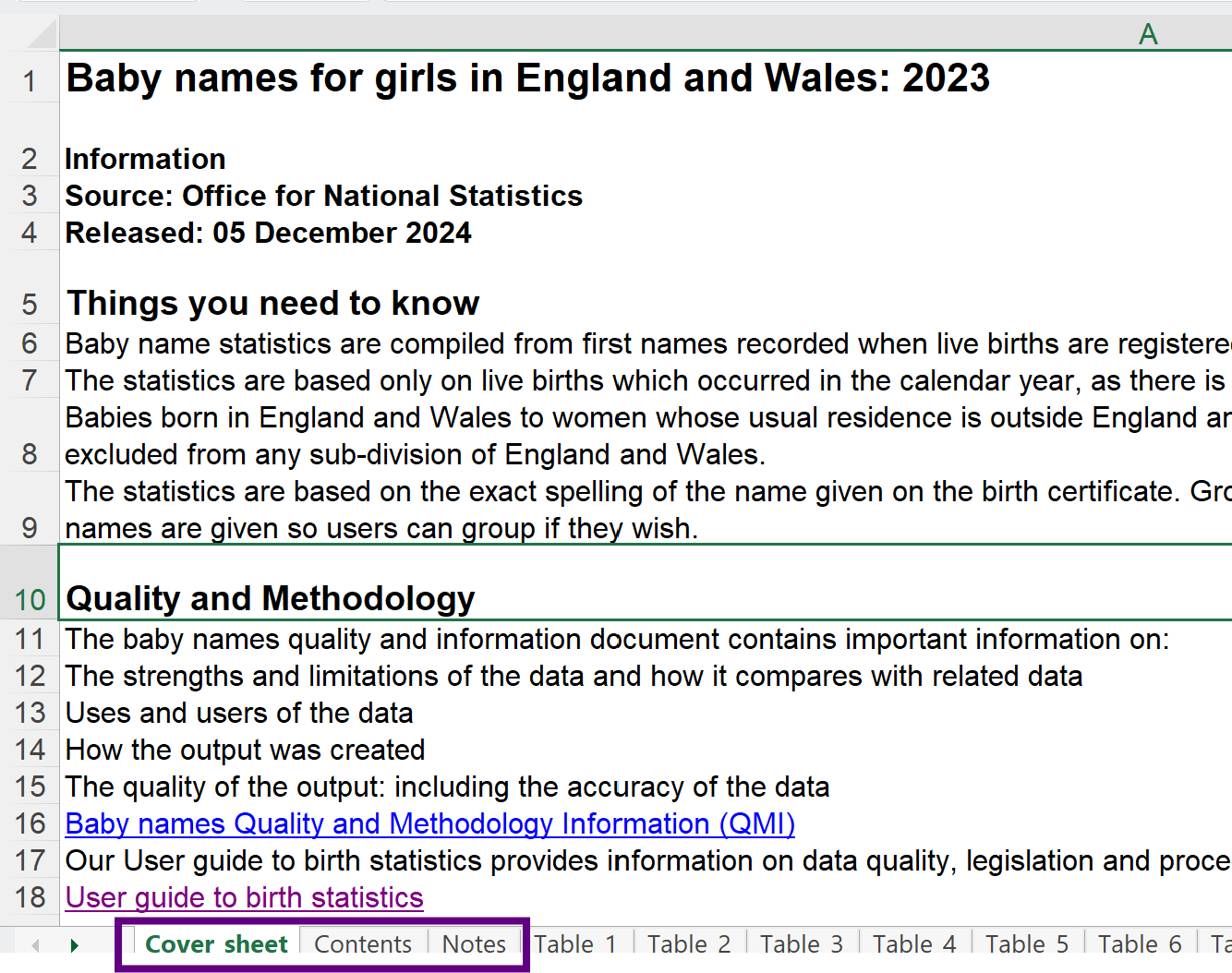
|  |  |
| --- | --- |
| ***Requirement Type*** | ***Requirement Description*** |
| Functional | The dashboard must allow users to filter names by region |
| Functional | There should be a bar chart that should display the top 10 names |
| Non-Functional | The dashboard must be easier to navigate and interactive |
| Non-Functional | All graphs and charts must include clear titles, labels and legends for interpretation |
| Functional | User can view top 10 Names by selecting an individual region (England & Wales- based regions). |
| Functional | The dashboard must include a line chart that shows how name popularity changes over months in 2023. |
| Non-Functional | All charts must use consistent fonts, colours, and spacing for a better appearance. |
| Non-Functional | Sections, Headers and any interactive elements should follow a consistent pattern to avoid user confusion. |
| Non-Functional | All pages within the dashboard should follow a common layout style. |
| Functional | An information panel should display the total number of babies registered. |

***Data Preparation***

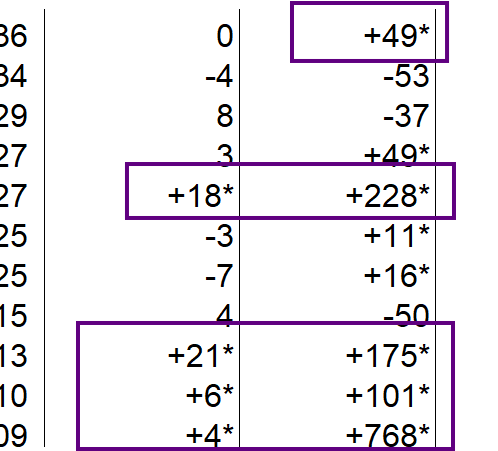
For data preparation, I handled the process manually since only a few adjustments were needed. I started by removing any unnecessary text that wasn’t relevant to the dashboard.



Also, some pages in the dataset were removed as they only contained source information, which was not needed for the visualisation. This can be seen in the screenshot below:



Secondly, I removed the asterisk symbols, as they would have interfered with the creation of graphs and charts.



Lastly, I copied and pasted the data into a new spreadsheet using values only to remove any hidden formatting or formulas that could cause issues when visualising the data in the dashboard.

Now the dataset is prepared to be used in the dashboard.

This is the cleaned version for the baby names of girls in England & Wales: [GirlsNamesDataset.xlsx](Data_Files/GirlsNamesDataset.xlsx)

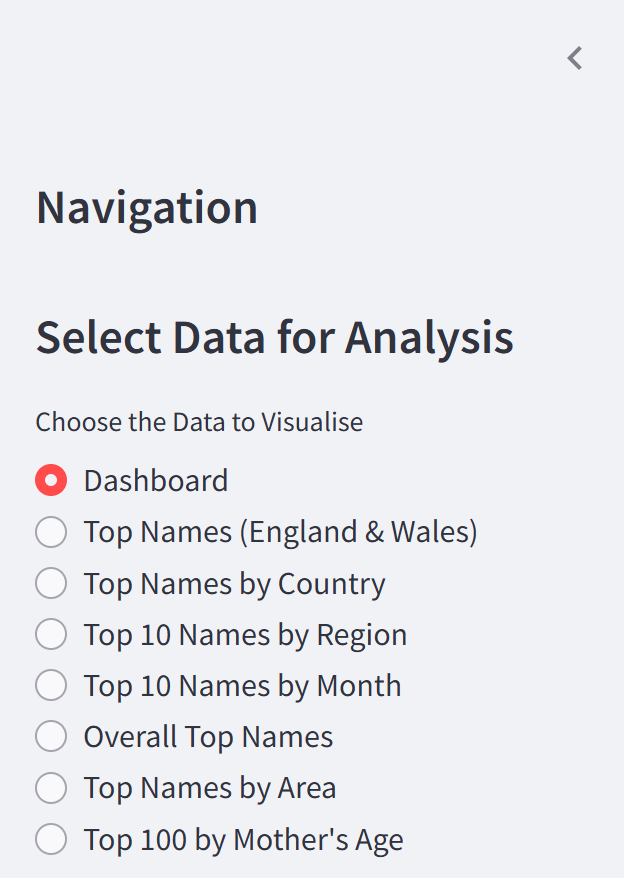
***Streamlit Dashboard***

The dashboard is creating using Streamlit to visualise data from the GirlsNamesDataset file. The goal was to make the data accessible and visually insightful, allowing users to filter, view, and understand trends in girl’s names over time. The dashboard was deployed online for the purpose of easily accessing and sharing.

The coding language used is python, and the code mainly utilised if statements that respond to user input from the Streamlit sidebar (Navigation).

One problem I faced was that the dataset contained 8 pages, rather than different excel files, to solve this problem, I then had the python file to read each sheet as a different data frame. This eventually helped solved the problem. Since there 8 pages in one dataset file, I decided to have a navigation section which will separate them to make it look more organised. The navigation button is on the top left side of the page.





Another problem I faced was the 8th page had different age groups, each of the age rank is divided into 4 age groups with their rank and name. To solve this problem, I created a list function which is grouping the columns in a specific format.

Once the dashboard was created, I had to upload it to streamlit website from GitHub. However, when I deployed it, it gave errors, such module not found error, files path errors, version mismatches. To solve these issues, I created a requirement file in a txt format and wrote all the necessary libraries in it. Also, added version numbers to that file to ensure compatibility and stability because some python packages required specific versions. Moreover, I also added a dependency like openpyxl in the requirements file because it is used for opening and reading excel since my code has pd.read\_excel and the chosen file was in excel format rather than csv. By adding that, it resolved the issue and the dashboard was successfully deployed and working.

Furthermore, to enhance the dashboard further, I would add advanced filtering options, and use functions rather than if statements to reduce the lines of code and location where the user can download the file if needed as an option.

***Test Cases***

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***TC1*** | ***Title***: Region Dropdown | ***Test Case Title***: Region Dropdown Functionality |
| ***Description*** | | Confirm that the region dropdown populates with all unique regions from the dataset. | |
| ***Steps and input data*** | | 1. Navigate to Top 10 Names by Region 2. Click on the Select Region Dropdown 3. Observe the options listed and click any region | |
| ***Dependencies*** | | df4 must be loaded with the correct “Geography” column | |
| ***Expected result*** | | Dropdown shows a full list of all region names e.g., North East, London, Wales | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***TC2*** | ***Title:*** Monthly Names | ***Test Case Title:*** Bar Chart for Monthly Names |
| ***Description*** | | Ensure that a bar chart correctly displays the top 10 names for a selected month. | |
| ***Steps and input data*** | | 1. Navigate to Top 10 Names by Month 2. Select ‘June’ from the month dropdown 3. Observe the bar chart | |
| ***Dependencies*** | | df5 must contain data for June  Data must exclude “All Names” | |
| ***Expected result*** | | A bar chart displaying the top 10 names only for a selected month | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***TC3*** | ***Title:*** Region Pie Chart | ***Test Case Title:*** Pie Chart shows Accurate Regional Distribution |
| ***Description*** | | Ensure that the pie chart correctly visualises the proportion of baby names registered for all regions. | |
| ***Steps and input data*** | | 1. Navigate to Top 10 Names by Region 2. Scroll to pie chart 3. Observe the pie chart | |
| ***Dependencies*** | | The region names in df4 must align with the labels shown in pie chart. | |
| ***Expected result*** | | The pie chart should show correct proportions; London has the largest slice, North East has one of the smallest. | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***TC4*** | ***Title:*** Monthly Line Chart | ***Test Case Title:*** Line Chart Incomplete |
| ***Description*** | | Check if the line chart shows the name trends across months includes a working legend to differentiate names. | |
| ***Steps and input data*** | | 1. Navigate to Top 10 Names by Month 2. Scroll to Line chart 3. Look for a legend showing the name each line represents | |
| ***Dependencies*** | | Each plotted line must have a label set  Use of legend to differentiate | |
| ***Expected result*** | | A completed legend with all 10 name labels should be visible, allowing the user to know which line is which. | |

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***TC5*** | ***Title:*** Top Names Mothers Age | ***Test Case Title:*** Mother’s Age Group filtering shows top names |
| ***Description*** | | Check that selecting an age group correctly filters the top names by rank and displays them in the bar chart. | |
| ***Steps and input data*** | | 1. Navigate to Top Names by Mother’s Age 2. Select Aged under 25 3. Choose Top 10 from the select the number of top names dropdown 4. Observe the bar chart | |
| ***Dependencies*** | | df8 contains correct column structure for each age group | |
| ***Expected result*** | | The chart shows 10 bars for the most popular names in the selected age group of mothers, sorted by ascending rank (1 being the most popular) | |

Table sheets from 1 to 8 are named in order of df1 to df8.

***Test Log***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TC** | **Date** | **Executed by** | **Actual Result** | **Pass/Fail** | **Notes** |
| 1 | 04/05/2025 | Tester | All regions displayed correctly | Pass | - |
| 2 | 04/05/2025 | Tester | Chart loads with the 10 bars for June | Pass | - |
| 3 | 04/05/2025 | Tester | Pie chart matched expected distribution | Pass | - |
| 4 | 04/05/2025 | Tester | Chart appeared but without its legend | Fail | ax.plot() is missing a label= argument |
| 5 | 04/05/2025 | Tester | Chart showed names sorted by rank | Pass | - |

By ‘Tester’, it means a family member.

This sign “-” means that there are comments/notes needed since the test was a pass.

To conclude, this dashboard brings baby name for girls’ data to life, offering an engaging way to explore how names vary across regions, months, and demographics. It highlights meaningful trends while keeping the user experience simple and interactive.