**Investigation**

I used a **flowchart** to organise my thoughts as I was reading the brief. The *most appealing ideas:*

* *Population changes*. Analysing population increase can highlight the solutions for housing crises and economic instability.
* *Third-level education.* Analysing trends in education since 2000s can highlight social progress and industry demand.
* *Wages in SMEs vs MNCs.* I identified a gap between wages offered by Small-Medium Enterprises and Multinational companies.

|  |  |  |  |
| --- | --- | --- | --- |
| Factor (out of 10) | Wage gap | Population | Education |
| Personal interest | 2 | 6 | 9 |
| Achievable | 4 | 4 | 2 |
| Variables *number of variables/grade* | 3/6 | 3/6 | <6/2 |
| Datasets available | 2 | 9 | 9 |
| Time consumption | 9 | 6 | 2 |
| Total grade | ***23*** | ***31*** | ***24*** |

After further research I **rejected ideas** with lower grades:

1. *Education:* considering all variables will be too difficult in the given timeframe.
2. *Wage gap*: strong idea, but lacking datasets.

I **chose Population** because:

* There are various datasets and examples available.
* Has only 3 variables and I’m confident in setting goals and meeting deadlines.
* I have clear idea of analysis to complete.
* I believe this area is important for society.

***End user*** will use my website fo**r:**

* *University students* (18-30 years) analysis and academic research for projects and dissertations.
* *Journalists* (25-50 years) analyse trends, enhance accuracy and fact-check their work.

I analysed various **interactive websites** starting with one listed in the brief: *worldbank* and then found some others: *worldometers* and *PRB.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Worldbank** | **worldometers** | **PRB** |
| **Focus** | Food prices/nutrition | Current population | Fertility rates |
| **Interactivity** | Animated charts and maps | Animated tables, modifiable charts (zoom in in specific decade) | Filter options, select location or tick boxes, search option |
| **Visualization** | Maps, bubble charts and colourful design | Maps, tables, line/pie charts | Projected charts, engaging layout |

**Evaluation:**

Worldometers and PRB relate to my chosen topic, and I’ll implement **features** that I **liked** from them: filter selector *(PRB)* and modifiable charts and layout *(worldometers)*.

I also **liked** colourful design in *worldbank*.

While I wouldn’t be able to implement all features due to time, **I aim** to provide user with clear understanding of population trends and potential shifts that may occur.

Features I **didn’t like**:

* Excessive amount of text which made the content harder to navigate (PRB).
* Excessive interactivity/movement which distracted and worsen user experience (worldometers)

Next, I investigated dipfferent **datasets** which would allow me to build my website.

**Population** and **land area** of *all countries over time* are important factors I need in my dataset

|  |  |  |  |
| --- | --- | --- | --- |
| Factor | Kaggle | World population review | World bank |
| All countries | ✓ | ✓ | ✓ |
| Land area | X | ✓ | X |
| Different years | ✓✓ | ✓ | ✓✓✓ |

There were **many datasets** available. Some of them I had to **reject** but I found one which had **all variables** – *World-population-review.*

**Plan and Design**

**My proposed solution:**

1. Display three interactive charts for density, country size variation, and world predicted population, **helping users** visualize global data, track changes and impacts of population.

2. Update graphs based on filters: show country’s density, EU/non-EU size variation, or predicted population for chosen years.

3. Store survey inputs in Firebase and display all users' statistics: average age, EU/non-EU percentage, and other options percentage.

This will **help user** to compare their choices with others and see the statistics for general knowledge.

4. On the recommendations page, ask for users' preferences and suggest 10 countries based on information from dataset.

This will **help users** who are willing to travel or move to other countries.

I’ll use **Extreme Programming Agile Framework** approach because it breaks large milestones into simple tasks. I’ll have continuous iterations to improve design and functionality.

**Wireframe**

**Basic requirements:**

|  |  |
| --- | --- |
| Collect and prepare data | I’ll use dataset from *World-population-review*, save it to csv file, clean it with pandas and store it in a dictionary. |
| Data analytics and visualisation | For my three graphs I’ll calculate density, standard deviation, predict population and measure percentage increase. I’ll use matplotlib to create line, bar and pie charts. |
| Create interactive information system | I’ll use JavaScript, HTML and CSS to create a website, and display 3 charts using Chart.js. |

**Advanced Requirements**

|  |  |
| --- | --- |
| Interaction | I’ll set selection areas, where user could choose which country to display. I’ll create checkboxes to tick and modifiable charts. |
| Form/poll | I’ll create a separate area where users could enter inputs. A summary of user inputs will display. |
| Recommendations | I’ll create a separate area where users could get recommendations based on their preferences. |

**Social/ethical/moral considerations:**

*Privacy:* No names or personal information will be displayed, only overall statistics.

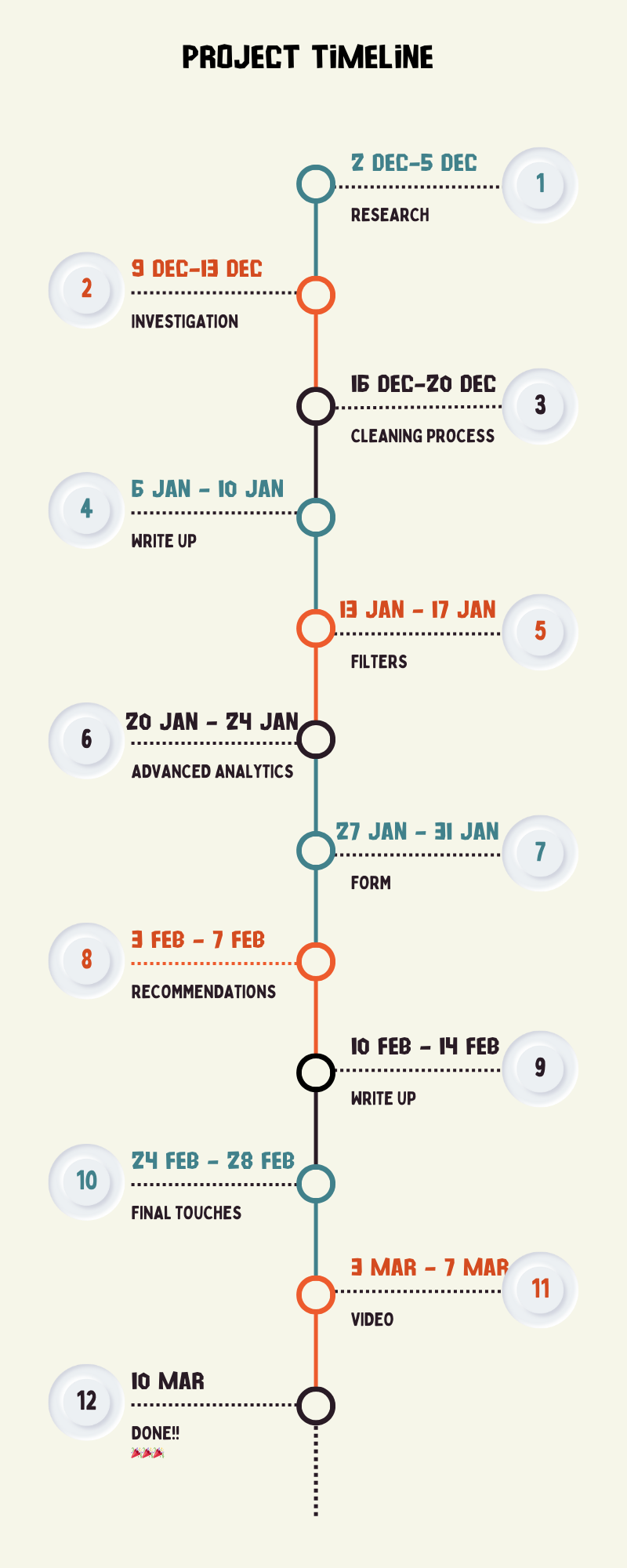
*Content Responsibility****:*** No harmful, biased, or misleading information will be promoted.

*Accessibility:* my website will be accessible by all, regardless of disabilities. It will have colourful charts, which are easy to read and simple interface.

|  |  |
| --- | --- |
| **Technology** | **Role** |
| Python | Cleaning, data analytics, visualisation and sending variables to database. |
| JavaScript | Visualising graphs, process user input and send it to database. |
| CSS | Establish alignment and design of website. |
| HTML | Create website’s structure. |
| Firebase | Store data analytics and user input. |
| Csv | Analyse dataset in csv file and use it for calculations. |
| miro.com | Create a flowchart and architecture diagram. |
| balsamiq.cloud | Create wireframe. It’s great for quick prototyping and design plan. |
| Thonny | It is easy to use and supports debugging. |
| VScode with live server | It supports HTML, JS, and CSS and has many extensions that speed up my work. |

**Flowchart**

**Create**   
  
**Timeline**



|  |  |
| --- | --- |
| Week | Progress: |
| 1 | * Identified 4 main ideas * Searched datasets and identified pros and cones * Searched other similar websites |
| 2 | * Picked population dataset * Plan the system work and analytics |
| 3 | * Used pandas to clean dataset * Stored clean data into dictionary and firebase * Created separate functions for analytics |
| 4 | * Completed report: “Investigation” & ”Plan&Design” |
| 5 | * Introduced filters on website * Saved inputs into variables |
| 6 | * Carried out analytics for filtering in python sent them to firebase * Used matplotlib to visualise basic graphs |
| 7 | * Created a form with questions * Stored validated inputs and send to firebase * Fetched all inputs from firebase * Displayed overall statistics |
| 8 | Created recommendations with options to select |
| 9 | * Completed report: “Create” & “Evaluation” |
| 10 | * Tested all graphs, fixed bugs * Checked inputs/output |
| 11 | * Recorded video |
| 12 | * Check everything was working and met all requirements of brief |

**Testing**

**Unit testing:**

Function which updates chart 2 when checkboxes are ticked and shows all countries by default.

Function gettingChart2() fetches data from firebase and calls updateChart2().



Current **test-table**:

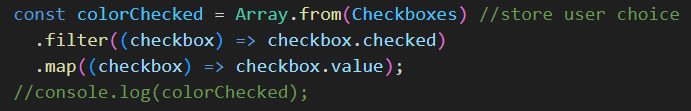
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test number** | **Checkboxes ticked** | **Expected output** | **Output** | **Test pass** |
| 1 | none | World’s variation | expected | Yes |
| 2 | EU | World’s+EU | World's only | No |
| 3 | Non-EU | World’s+non-EU | World's only | No |
| 4 | both | World’s+EU+non-EU | World's only | No |
| 5 | Untick | World’s variation | World's only | Yes |

There are multiple **issues**:

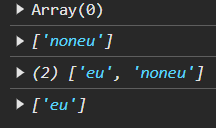
1. User’s choice is not stored/checked.
2. EU/non-EU aren’t shown according to user’s choice
3. Each bar in dataset doesn’t have a unique label/colour
4. Instead of creating a newChart, old one should be updated

**How I fixed it?**

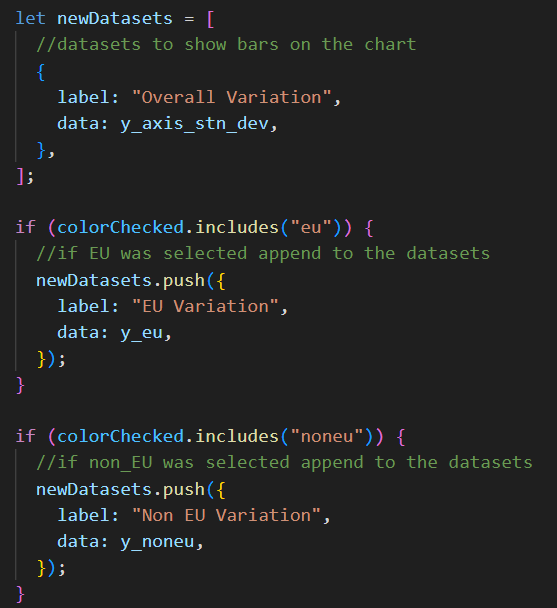
I created a variable which would store user choice:



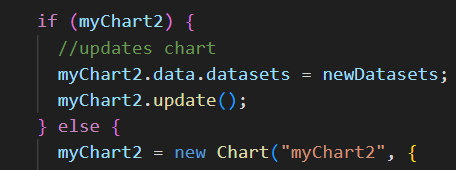
And checked its value using console.



I initialized object newDatasets to store data for updating chart. It has label and dataset for default chart. If EU/non-EU option was selected, it was added to object, ensuring when checkbox ticked, object updated.



I set an if/else condition to check if chart already exists:



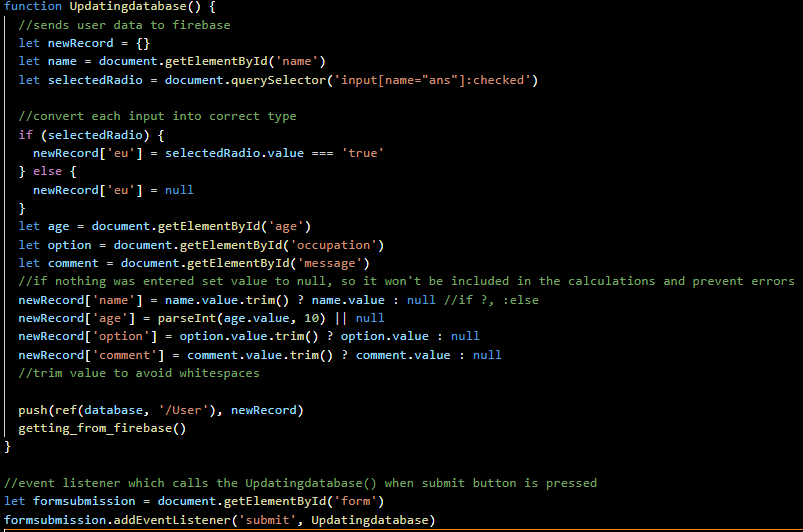
If it exists: old chart updated, otherwise new one created.

I combined final **test table** to ensure that all tests are passed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test number** | **Checkboxes ticked** | **Expected output** | **Output** | **Test pass** |
| 1 | none | World’s variation | expected | Yes |
| 2 | EU | World’s+EU | expected | Yes |
| 3 | Non-EU | World’s+non-EU | expected | Yes |
| 4 | both | World’s+EU+non-EU | expected | Yes |
| 5 | Untick | World’s variation | World's only | Yes |

**Regression testing:**

Adding validation in the form on the age input and making only numbers acceptable or no value.

  
Current function accepts text input in age field and saves it as null.

Current **test table**:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test number** | **Input** | **Expected output** | **Output** | **Correctly updated results** | **Test pass** |
| 1 | “26” | accept | accept | yes | Yes |
| 2 | “0” | Invalid\_input | accept | No | No |
| 3 | “-54” | Invalid\_input | accept | No | No |
| 4 | “5.4” | Invalid\_input | accept | No | No |
| 5 | “132” | accept | accept | yes | yes |
| 6 | “ten” | Invalid\_input | accept | No updation | No |
| 7 | “false” | Invalid\_input | accept | No updation | No |
| 8 | “Meow” | Invalid\_input | accept | No updation | No |
| 9 | None | accept | accept | yes | yes |
| 10 | [23,54,32] | Invalid\_input | accept | No updation | No |
| 11 | “456” | Invalid\_input | accept | No updation | No |

**Impact of test table?**

* Artefact accepts all input, even those that should be marked invalid.
* Expected outcome was to reject non-numeric, negative or float inputs.
* Invalid inputs didn't update results.
* System should reject all invalid inputs.

**Major problem1:**

*System accepts all inputs.*

**How I approached this?**

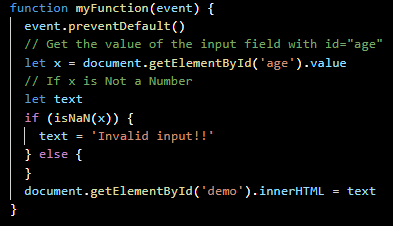
I added a paragraph tag to show errors to users, if NaN was entered.



I set an onclick feature to submit button.



I created a function which would show appropriate message to the user:

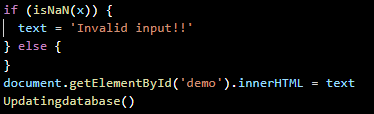


However, I faced with a problem of onclick feature causing ReferenceError, so I replaced it with eventListeners().



This resulted in Updatingdatabase() not being called due to 2 eventListeners().

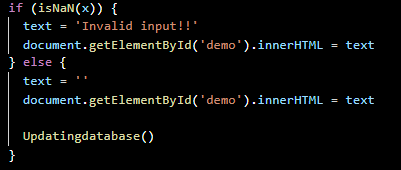
Therefore, I deleted Updatingdatabase eventListener and put its function call inside of myFunction().



There were no errors, however when a valid input was entered, unwanted message appeared.



I set text to “” if valid input was entered.

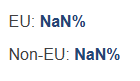


I added more conditions for validation:



**Major problem2:**

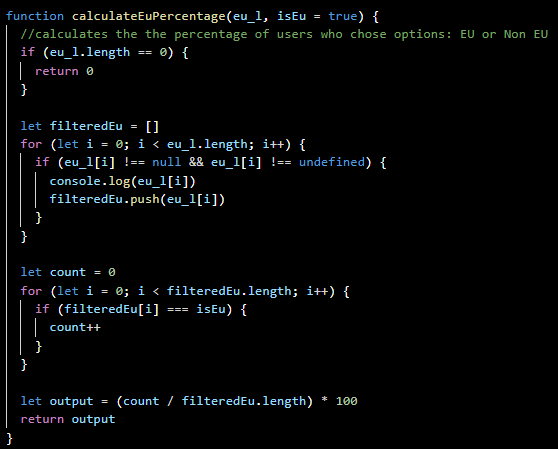
I got **NaN% in “Results”** when radio was not filled in.



Such output could occur due to division by 0:

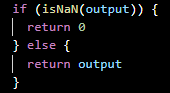


In this function:

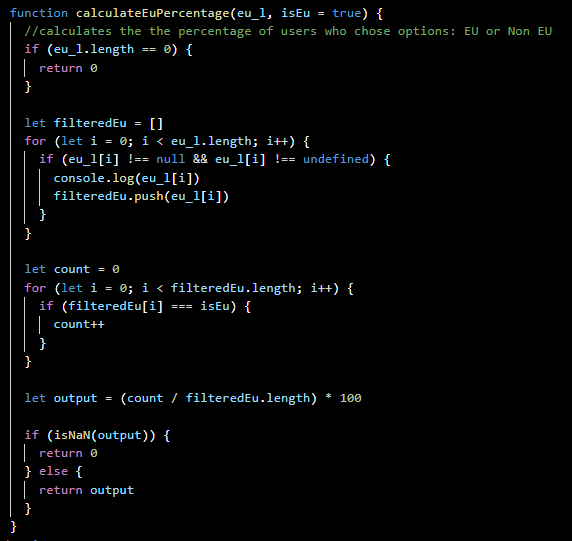


I handled this issue before by specifying if length of list is 0, return 0%.

NaN% should occur only when there is no previous answer submitted, I specified the return to be 0 if output is NaN.



Final function looked like:

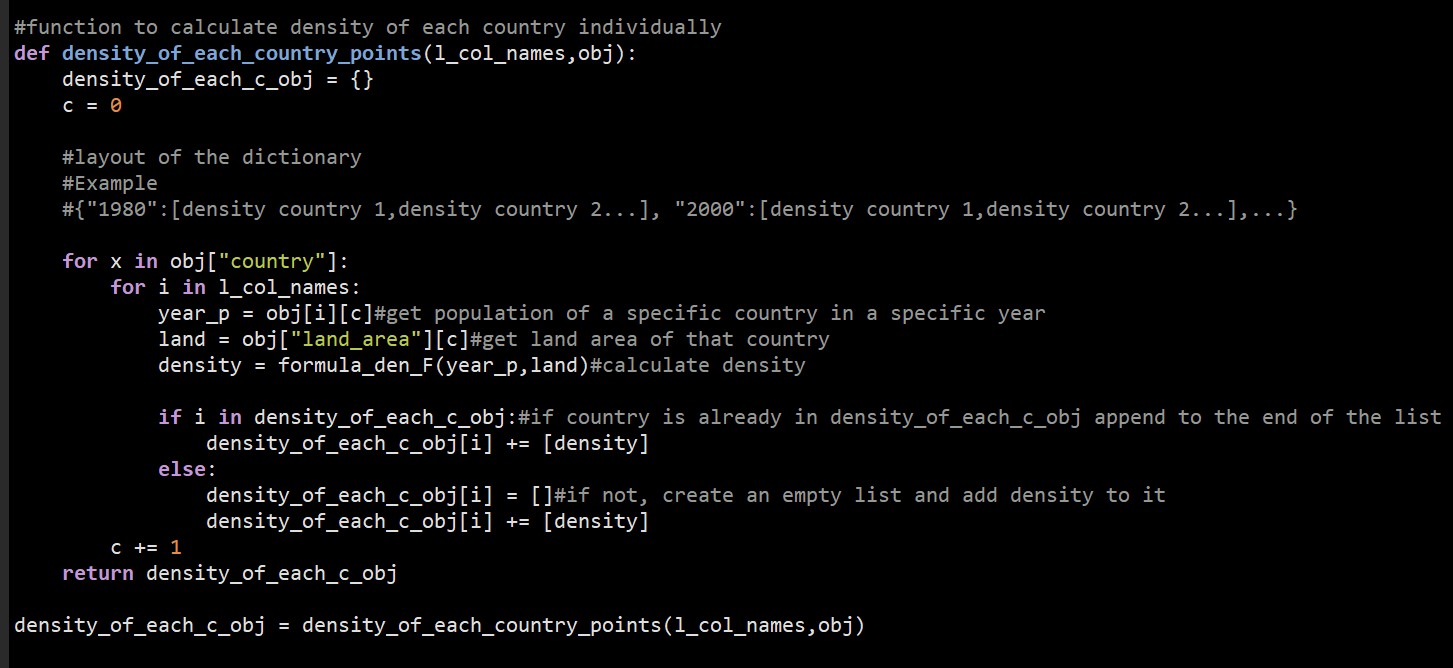


**Final test table:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test number** | **Input** | **Expected output** | **Output** | **Correctly updated results** | **Test pass** |
| 1 | “7” | accept | accept | yes | Yes |
| 2 | “0” | Invalid\_input | Invalid\_input | No updation | yes |
| 3 | “-18” | Invalid\_input | Invalid\_input | No updation | yes |
| 4 | “12.8” | Invalid\_input | Invalid\_input | No updation | yes |
| 5 | “176” | accept | accept | yes | yes |
| 6 | “five” | Invalid\_input | Invalid\_input | No updation | yes |
| 7 | “true” | Invalid\_input | Invalid\_input | No updation | yes |
| 8 | “Hi” | Invalid\_input | Invalid\_input | No updation | yes |
| 9 | None | accept | accept | yes | yes |
| 10 | [14,67,98] | Invalid\_input | Invalid\_input | No updation | yes |
| 11 | “256” | Invalid\_input | Invalid\_input | No updation | yes |

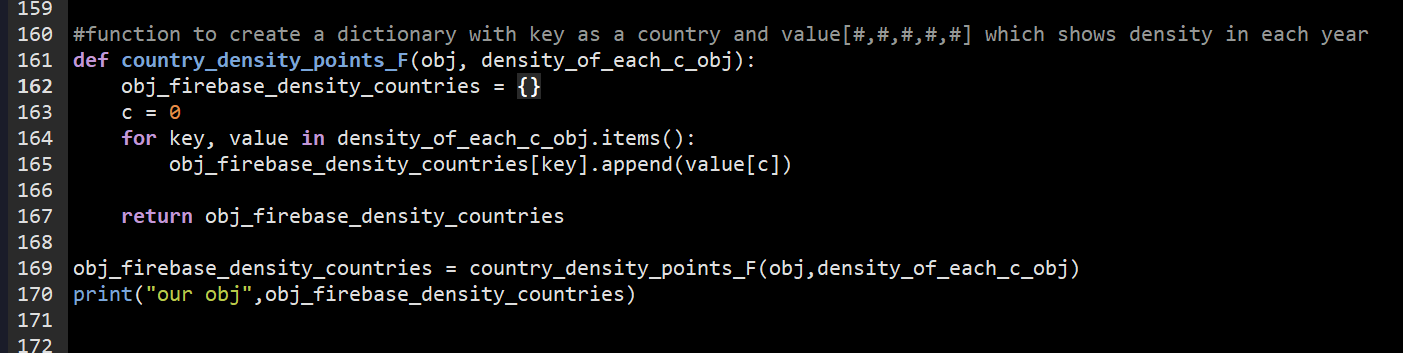
**All tests passed!**

**Problems**  
  
**Major problem** for me was to format each countries’ density into dictionary without breaking the order.



Challenge was to map densities to their respective countries.

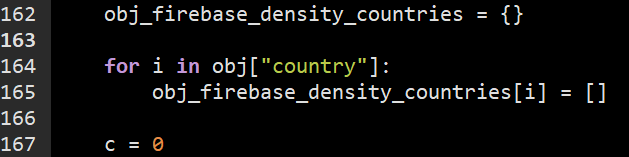
I used a separate function:

When it was run, I got an error on line 165:

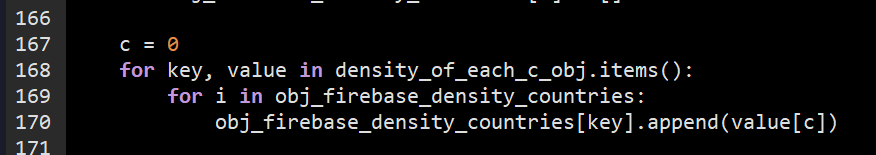


KeyError helped me to understand the problem, as each key in obj\_firebase\_density\_countries should be a country, not a year.

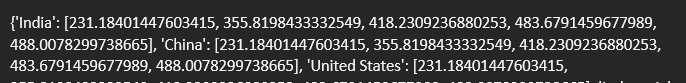
I added for loop to set each country as a key:



Also, added another for loop to go through each key and append respective density:

I replaced [key] with [i], as ”key" is a key in density\_of\_each\_c\_obj:

Then I realised that each country had the same density:

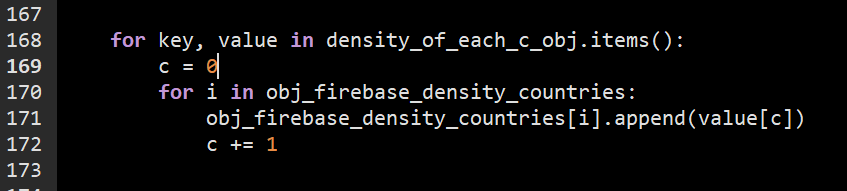


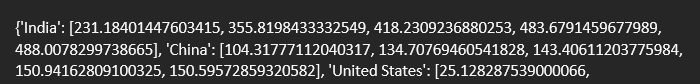
This is because c is a constant, so I added 1 to it in the loop:



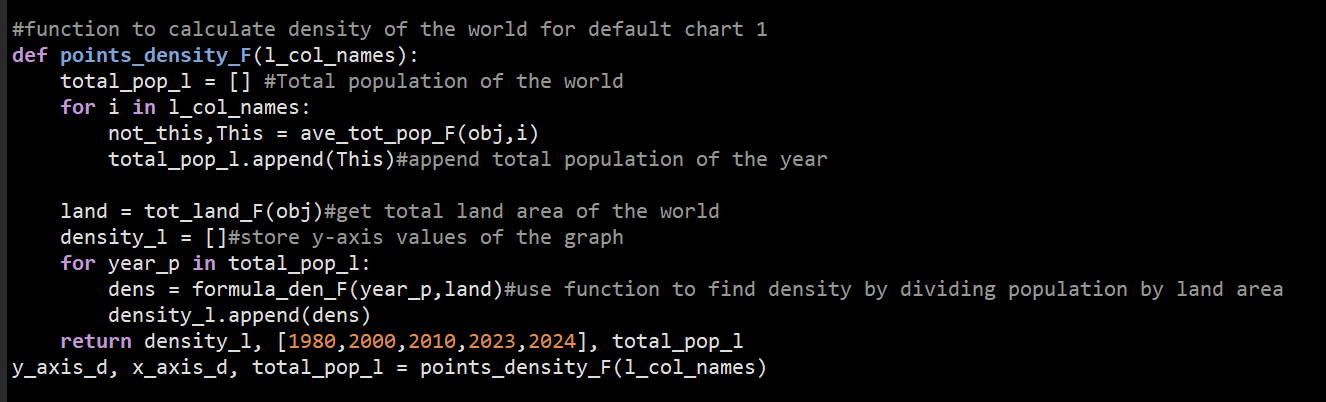
After next run I got an indexError:

c wasn’t updated in the first loop, increasing from its initial value as loop processes new values. Moving it inside the loop fixed this.

The final output was in the correct format:



**Code explanation/pseudocode for analysis of co-ordinates for chart 1:**



Initiate list total\_pop\_l to store total population of the world during 5 years.



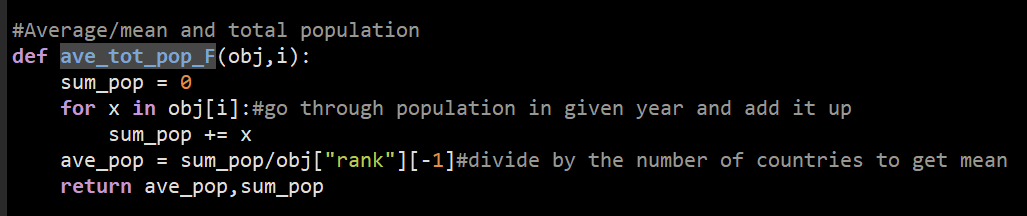
For loop goes through list of keys, which has only populations as values of all countries.



Function call ave\_tot\_pop\_F has 2 returns, but we need only second one - “This”.



Called function calculates total population of all countries in given year ”i”.

For loop goes through each value in dictionary with key “i” from l\_col\_names, which was passed as an argument in function call.



Each countries’ population is added up to form total population.



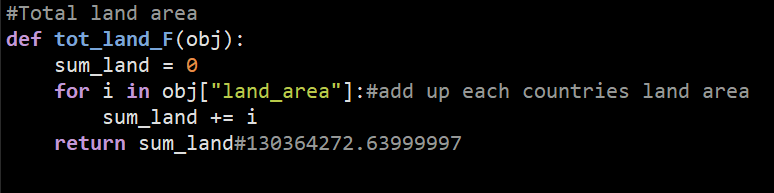
Coming back to original function total population appends to total\_pop\_l list.



Function tot\_land\_F() is called to calculate total land area of all countries. It is the same in each year, so no for loop needed.



Function goes through each countries’ land area and adds it to total area in sum\_land variable.

The return value in stored in variable land.

I initiated list density\_l to store y-axis co-ordinates of graph1.



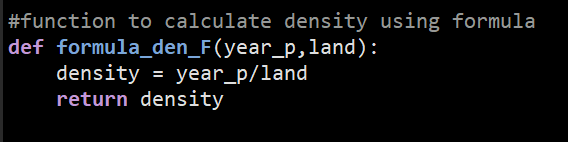
For loop then goes through each value in total\_pop\_l.



Function formula\_den\_F() is called storing return value in variable dens.



Function divides population by land area to get density.



Return is appended into y-axis list.



Original function returns x-axis and y-axis of the graph to plot, as well as total\_pop\_l for further use in other calculations.



**Why this approached?**

• Follows structured process, easier debugging.

• Separate functions: break the problem, reducing errors increasing readability.

• For-loops go through each country, ensuring all countries accounted.

**Other approaches:**

1. Single loop calculating density instead of separate functions.
2. Dictionaries instead of lists.

**Why weren’t used?**

Difficulty in modifying functions, harder debugging, reduced flexibility if using dictionaries, less reusability.

**Evaluation**

|  |  |  |
| --- | --- | --- |
| Requirements | How I did it? | Met/not met |
| **BR1.**Collect and prepare data | I found my dataset on *World-population-review* and downloaded as csv  I cleaned data using pandas and stored in firebase | **Yes!** |
| **BR2.**Data analytics and visualisation | I calculated density, standard deviation, predicted population and found %increase.  I used matplotlib for three graphs representing analysis. | **Yes!** |
| **BR3**.Create basic website | I used HTML, JavaScript, CSS and ChartJS to visualise charts on the website. | **Yes!** |
| **AR1**.Graph Interactivity | I used dropdown menus and checkboxes and ensured graphs update.  I used firebase to fetch data on the website. | **Yes!** |
| **AR2.**Form | I created a page with questions and showed results of all users.  I ensured validation on the inputs. | **Yes!** |
| **AR3.**Recommendations | I created a page with select options to recommend 10 countries.  I selected the displayed countries using score system. | **Yes!** |

If I had more time I would make the following improvements:

**Improvement 1:**

Create filter for chart1 which would sort countries’ densities in ascending order and display multiple countries on one chart using different lines.

This would make chart more flexible and informative, enhancing user’s experience. Ascending order will help to identify patterns and see countries with highest/lowest densities and better visualize data.

**Improvement 2:**

Ask more questions on recommendations page and suggest greater range of countries.

This would enhance my understanding of user preferences, resulting in better quality and accurate response.

**Other application for my artefact:**

Comment posting and replies.

On my form page, users can input comments, and my system could be adapted to display these publicly, allowing interactions, instead of just storing feedback. Such platform would create discussions, feedback exchange and real-time conversations in the community.

References:

1. <https://www.worldbank.org/en/programs/icp/brief/foodpricesfornutrition>
2. [https://www.worldometers.info/world-population/#google\_vignette](https://www.worldometers.info/world-population/" \l "google_vignette);
3. <https://www.prb.org/resources/understanding-population-projections-assumptions-behind-the-numbers/>