Link to repository: <https://github.com/csmfduro/salaries>

**Group Project Report: Data Analysis on AI/ML Job Salaries**

* **Open Data Considered by the Group and the Rational for the Final  
  S­­­­election (By the whole Group):**

After careful discussion among group members, we shortlisted three datasets from different categories to explore for this project:

* AI/ML Job Salaries
* Vehicles Dataset
* Genshin Impact Characters Dataset

We thought it would be interesting to consider datasets from different categories like professionalism, technical and entertainment. While each of the datasets were interesting to talk about. We eventually picked the AI/ML Job Salaries for the reasons listed below.

* It felt like the most relevant and professional option among the three datasets
* The dataset had a lot of valuable and interesting things to talk about like experience level, job type, remote work percentage, salaries and company size. This gave us a lot of things to extract and discuss from the dataset
* **Assessment of the potential insights that might be extracted from the data set (By the whole group)**

In total, we were able to extract a total of 14 interesting things to discuss on the dataset. Below will be the names of each group member and two(2) of their interesting insights:

**Fiyin:**

* Which countries offer the highest salaries for AI professionals?
* What are the highest-paying AI job titles?

**Prajesh:**

* How much would an Entry Level/Junior of a high paying job make compared to a low paying job but at senior level or Executive level?
* Do freelancers make more money or less money compared to employees who work fulltime and if so what job/ task allows a freelancer to get a good pay and if less what jobs are freelancers taking up?

**Jason**

* what is the difference in average salary between companies with 100% remote work and 0% remote work
* what is the difference between the average salary in lockdown (2020) compared to this year (2025)

**Jake**

* What are the job titles for the ten lowest paying jobs (in ascending order)?
* What is the main trend between experience level and salary?

**Faris**

* How do salaries differ between employees with and without a degree?
* Is there a correlation between years of experience and salary?

**Andrei**

* What is the average salary difference between the small, medium and large companies?
* What location of companies have the highest salary?
* **About Dataset**

The salaries are from ai-jobs. Ai-jobs collects salary information anonymously from professionals all over the world in the AI/ML and Big Data space and makes it publicly available for anyone to use, share and play around with. The data is being updated regularly with new data coming in, usually on a weekly basis.

The primary goal is to have data that can provide better guidance regarding what's being paid globally. So, newbies, experienced pros, hiring managers, recruiters and startup founders or people wanting to make a career switch can make better informed decisions.

The dataset contains one table structured as follow:

1. **work\_year**: The year the salary was paid.

2. **experience\_level**: The experience level in the job during the year with the following possible values:

EN: Entry-level / Junior

MI: Mid-level / Intermediate

SE: Senior-level / Expert

EX: Executive-level / Director

3. **employment\_type**: The type of employment for the role:

PT: Part-time

FT: Full-time

CT: Contract

FL: Freelance

4. **job\_title**: The role worked in during the year.

5. **salary**: The total gross salary amount paid.

6. **salary\_currency**: The currency of the salary paid as an ISO 4217 currency code.

7. **salary\_in\_usd**: The salary in USD (FX rate divided by avg. USD rate for the respective year via fxdata.foorilla.com).

8. **employee\_residence**: Employee's primary country of residence in during the work year as an ISO 3166 country code.

9. **remote\_ratio**: The overall amount of work done remotely, possible values are as follows:

0: No remote work (less than 20%)

50: Partially remote

100: Fully remote (more than 80%)

10. **company\_location**: The country of the employer's main office or contracting branch as an ISO 3166 country code.

11. **company\_size**: The average number of people that worked for the company during the year:

S: less than 50 employees (small)

M: 50 to 250 employees (medium)

L: more than 250 employees (large)

The reason we chose this dataset is that it had the most relevance with all the members of the group being about AI job salaries. When compared to our other options being cars and cancer probabilities it also was the most appropriate and containing the most data to be able to extract assumptions from.

* **Algorithms and Optimizations**

We used the following algorithms and techniques to effectively work on the dataset:

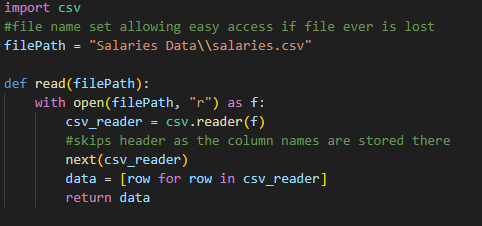
* **groupby() and mean calculation:** This technique involves grouping the data by specific columns, such as ***country*** or ***job\_title***, and calculating the average salary for each group. For example, to determine which countries offer the highest salaries for AI professionals, we group the data based on ***employee\_residence*** and calculate the average salary for each country. This algorithm is very efficient and effective because it reduces the redundancy of the code.
* **Sorting:** This algorithm is one of the most common in data analysis. It is simple but yet, effective. As the name suggests, it sorts out aggregated values based on the order you want them to appear. In our analysis, after aggregating data (e.g., average salaries by job title), we used sorting to arrange the results in descending order. This allows us to quickly identify the highest-paying roles or countries.
* **Visualizations:** Visualizing data is one of the most effective ways to present and understand trends. Using ***matplotlib***, we created bar charts to highlight the top-paying countries and job titles. Visual representations make it easier to spot patterns and communicate the findings clearly.
* The use of **.round(2)** for precision.
* **Application Design**

Our application’s functionality will include a main menu as we will not display the results for every question at once allowing the user to choose which questions they would like the data from. The menu options included are the ability to view the dataset, view the column names, view the dataset info and choose the questions to view the info from. The program will interact with the user through the user inputting the menu option they want to view. The data will be processed through csv and pandas to extract the dataset from the csv file.

**Implementation**

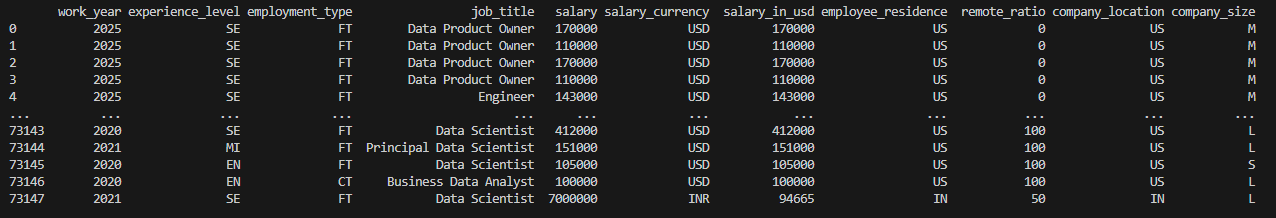
* **File reading function – Jason & Fiyin**

To begin with Jason created a reusable function to read the csv file and extract the dataset as this would be used for all the questions and main menu functions. This went through a few iterations as originally it used csv but he found using pandas was more streamlines and easy to understand

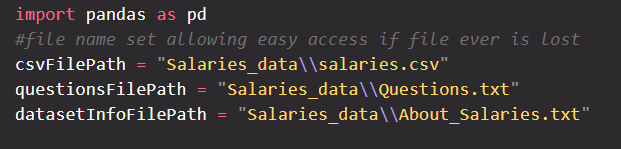


This is the original function to be able to read the dataset file which uses a variable “filepath” that can be easily changed at the top of the code in case the file ever gets lost or renamed. Using this the code opens the file and inputs the contents into a list and returns it to a variable. This ended up looking messy when viewed so I switched the code to use the pandas module instead.

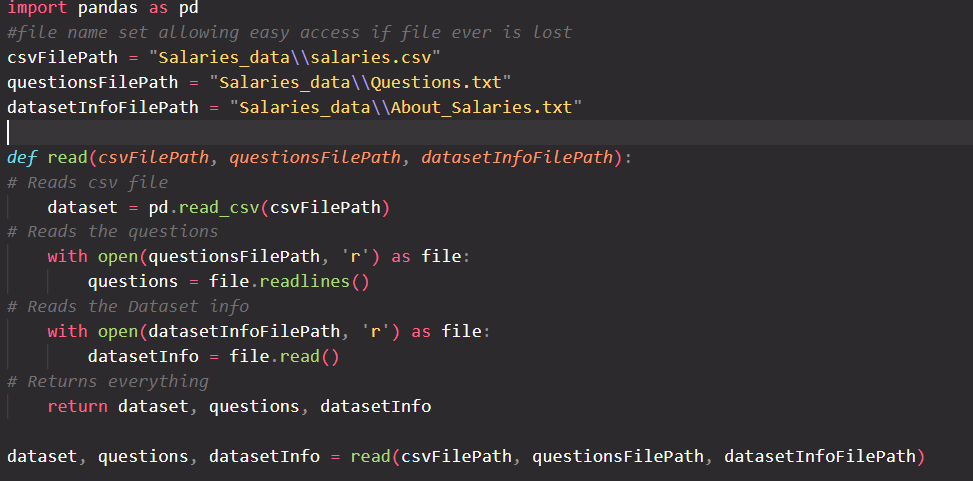


This is the final file reading function using the pandas module as the data outputted is automatically formatted making it easy to read as shown below.

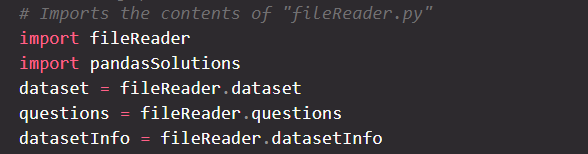
In addition to what Jason did, Fiyin further optimized the code by creating a separate file that reads all the readable data like the Path to the csv file, Path to the Questions file and the path to the dataset info.



Afterwards, Fiyin further added a function that reads all those files at the same. This made the reading of the files a lot easier for majority of the group members.

****

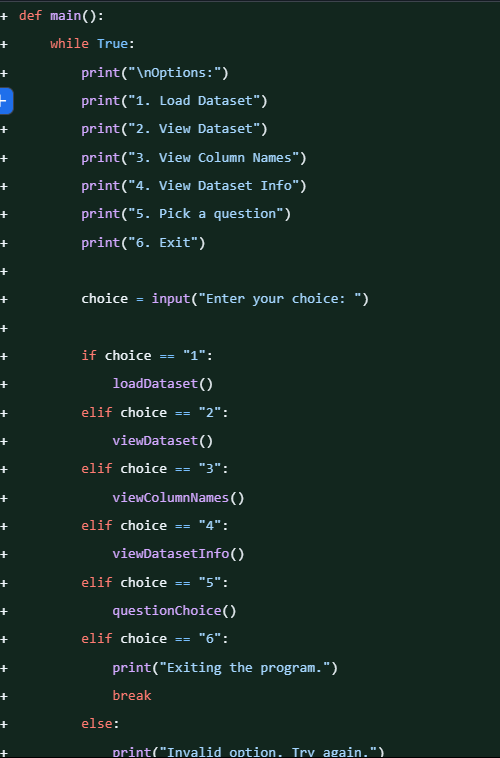
To use any of the files, all they’ll have to do is to import the module that reads all the files and proceed with the rest of their work.



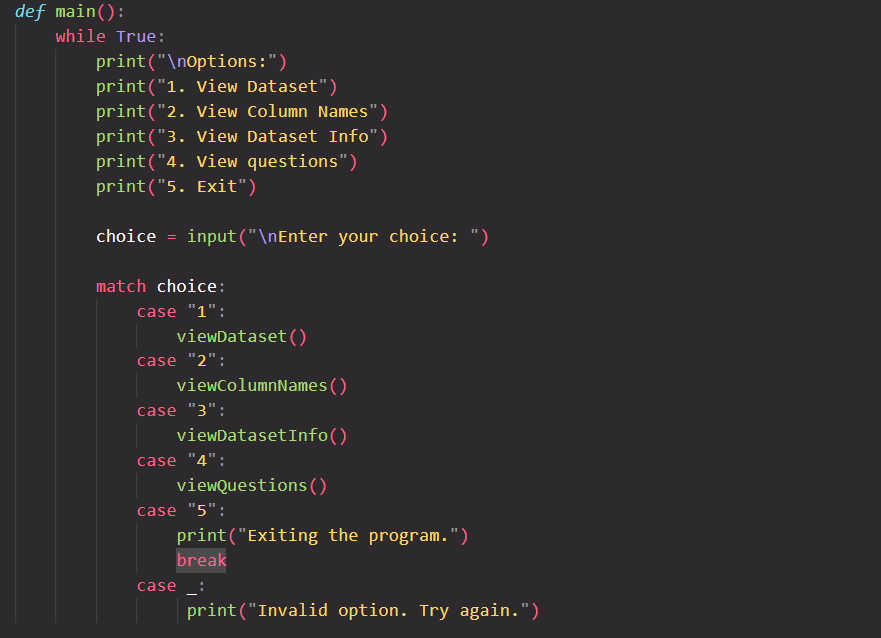
With just 4 lines, a lot of unnecessary time was saved

* **Main menu (structure) – Andrei, Prajesh & Fiyin**

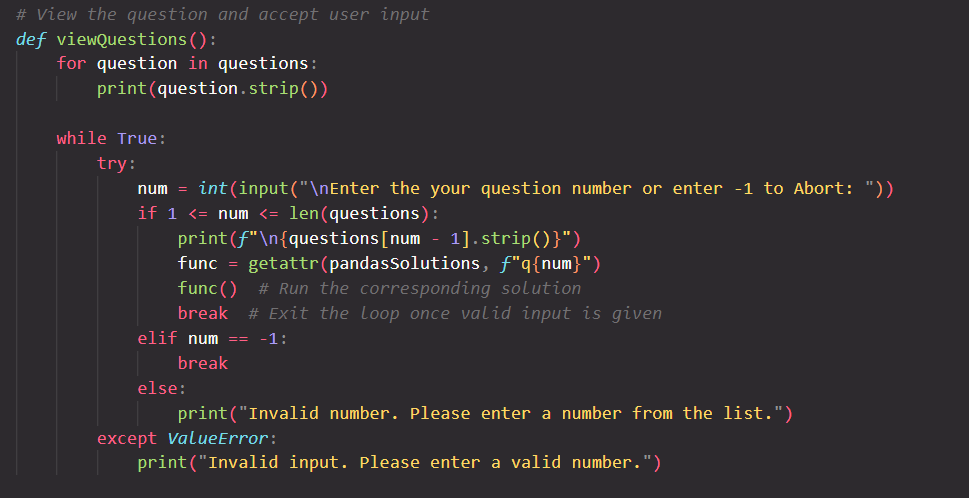
To create the main menu, Andrei and Prajesh listed out the possible options that the user would like to see, created a variable that stores the user’s input. And also created an if statement to run certain functions based on the user’s choice. The menu also checks if the value entered by the user is in the options. If it isn’t, they are prompted again till they enter a valid input. This loop continues until the user enters (5) which is the option for exiting the program.



Fiyin modified the menu and replaced the if-else statement with match-case statements. This made the menu more effective and more professional.



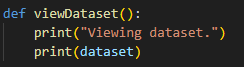
Based on the user’s choice, certain functions were called. One of the functions which was created by Fiyin, is called when the user wants to view the available questions on the dataset. The function starts by asking the user to enter the number that they are interested in. After the number has been entered, the function fetches the question and the solution to that question from a file called pandasSolutions.py. The file contains the solution to all the questions. After the solution is extracted, it is then displayed for the user to see.



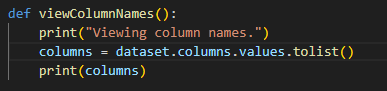
* **Main menu (function) – Jason**

Once the menu structure was finished including the shells of each function that would be used as menu options, I added the code to make them work. The options that were given to me are: load dataset, view dataset, view column names, view dataset info and choose questions of which I added the code for the first 4.

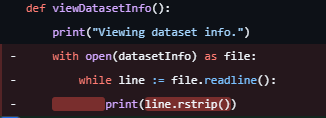
Starting with the load dataset function I thought that it made more sense to load it upon running the application so this option has been removed in favour for doing it upon program start.



The code to view the dataset is very simple as the bulk of the work was already done when creating the function to read the file as this function just prints out the info that was extracted from the file. This code went through no changes



The view column names function takes the dataset extracted using the file reading function and displays the top row of values being the column names. This code went through no changes.



The view dataset function went through a couple changes as this is the original. It read the file every time the user called the function which isn’t ideal, this was later changed by Fiyin.

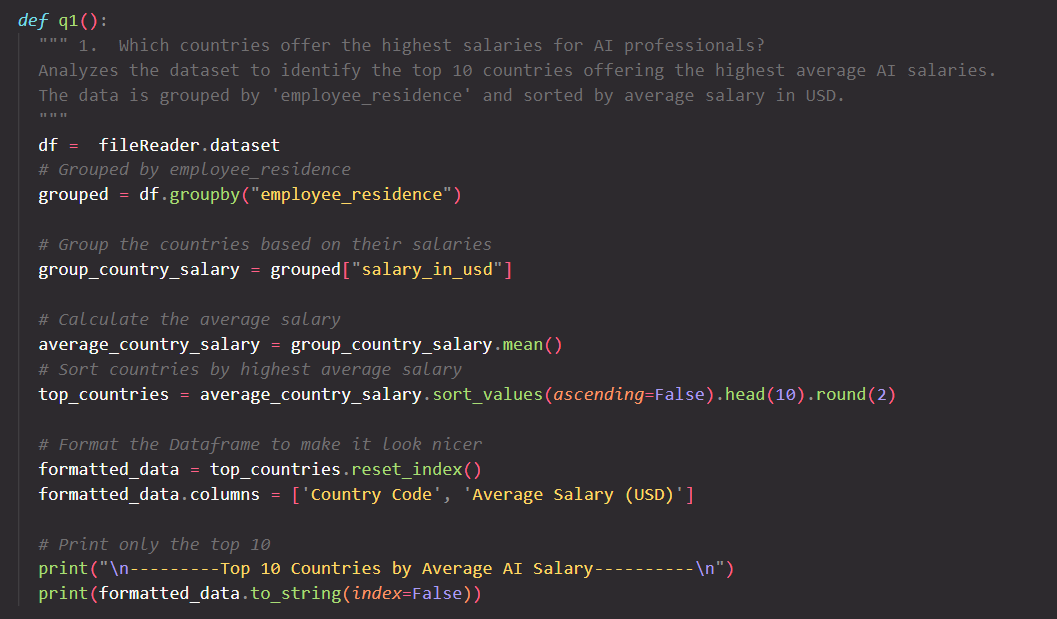
* **Group member’s questions and Report**
* Questions – Fiyin

Questions presented: 1 and 2

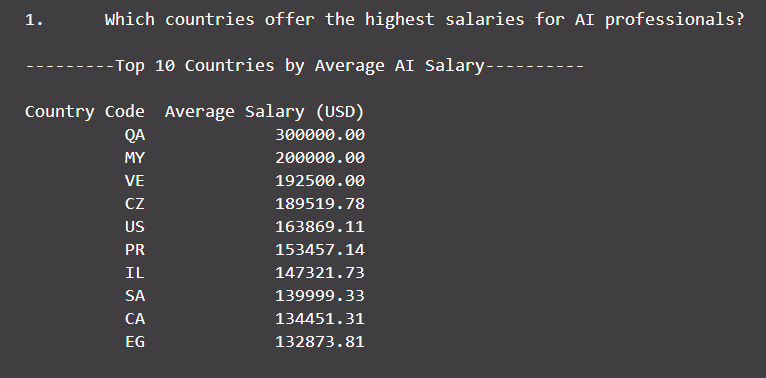
1. **Which countries offer the highest salaries for AI professionals?**

The first step I took in answering this question was to search for the relevant data in the dataset. Since I am looking for which **countries*(employee\_residence)*** offer the **highest salaries**, I am pretty much interested in only that for this question. Next, I grouped the dataset by the employee residence to represent the country of the employee. Afterwards, I further grouped the grouped data by the salary. After that, I found the average salary based on their country, sorted it, and printed it to the console. Once I had the average salary per country, I sorted the results in descending order and selected the top 10 highest paying countries. I reset the index of the result and renamed the columns to make the output more presentable.

Finally, I printed out the formatted results.



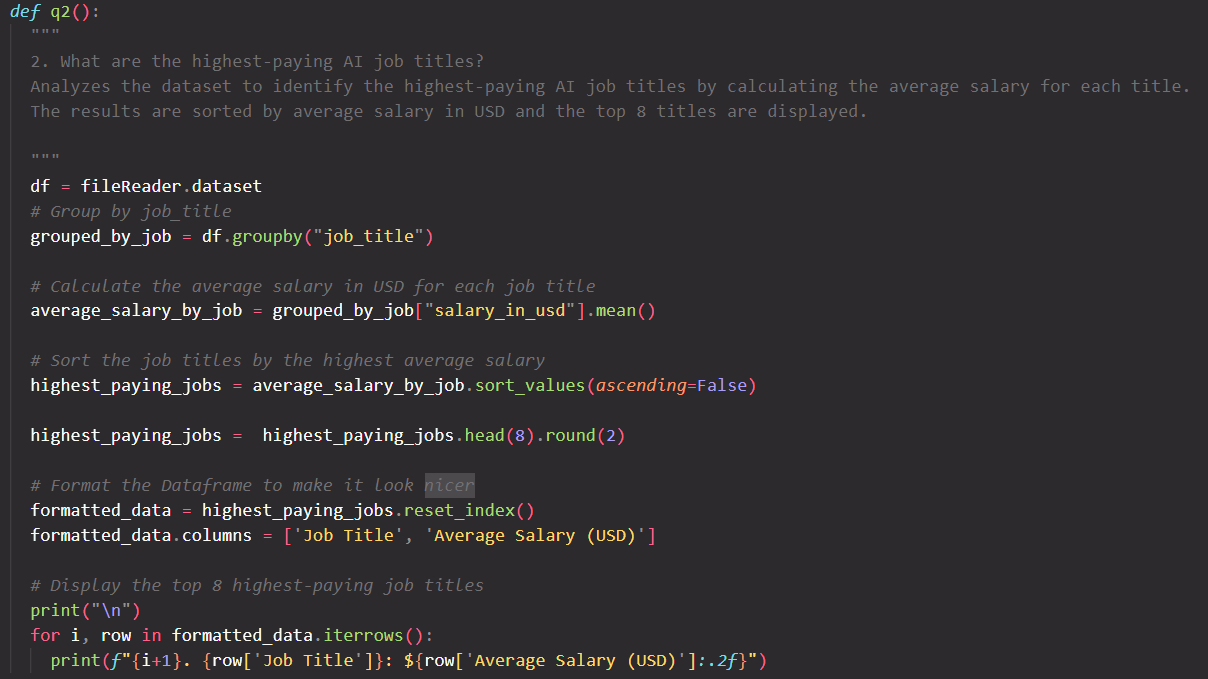
**Output**:



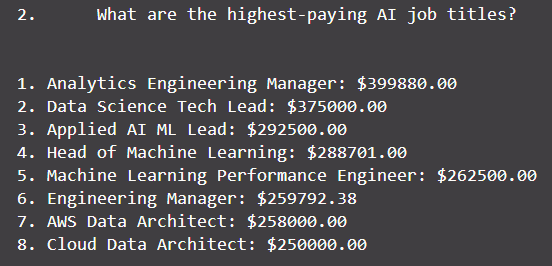
From this dataset, it shows that Qatar, Morocco and Venezuela pay the most for AI/ML Software engineers

1. **What are the highest-paying AI job titles?**

Somewhat related to question 1. This time, I analysed the dataset to identify the highest-paying AI job titles. To achieve this, I grouped the data by job\_title and calculated the average salary in USD for each title. After getting the averages, I sorted them in descending order to identify the top-paying jobs. I limited the output to the top 8 highest average salaries and rounded the values for clearer presentation. I added the last output with renamed columns to make it cleaner and more readable



**Output:**

****

Based on the dataset, the best and most proficient AI/ML Job in the world is An Analytics Engineering Manager with a salary of $399880

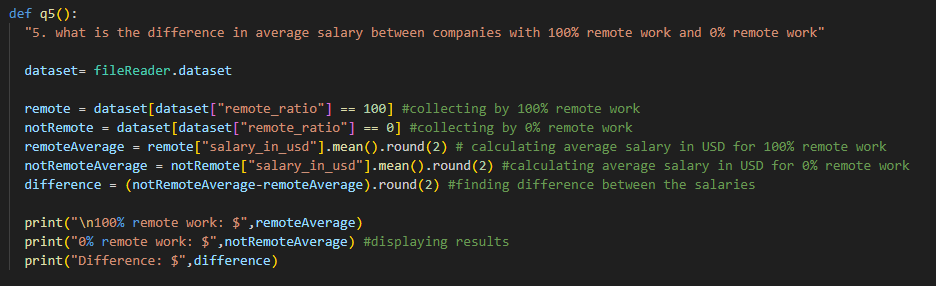
* **Questions – Jason**

My questions that I had to represent were

**5. what is the difference in average salary between companies with 100% remote work and 0% remote work**

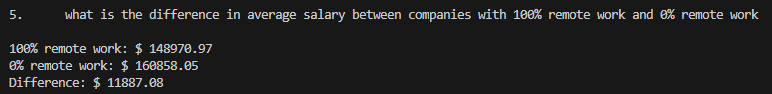
Both questions required similar code to calculate so there will be clear similarities when showing both pieces of code used for each.

The code for the first question being the difference between full remote work and no remote work first retrieves the file from the dataset then extracts both the positions with 100% remote work and 0% remote work and calculates the mean of each respective salary. Finally, the code calculates the difference between the 2 averages before printing the results out in a clear way.

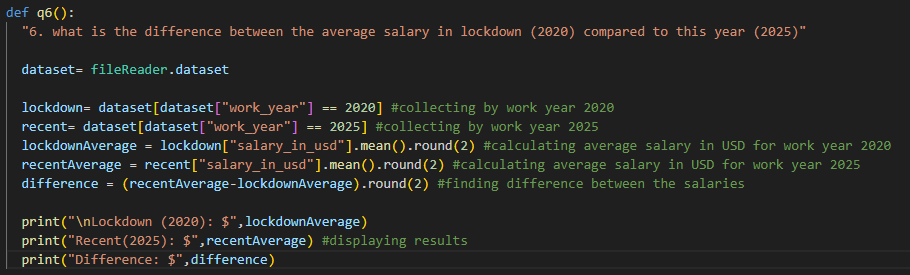


The result of the question is that on average jobs with no remote work pay slightly more than jobs that have 100% remote work.

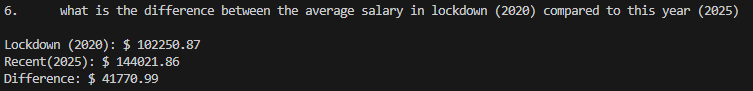
The difference is very minimal.

The output of the above code   


**6. what is the difference between the average salary in lockdown (2020) compared to this year (2025)**



This code for this question works in a very similar manner to the other question substituting the remote work percentage for the work year comparing specifically 2020 to 2025.



The result shows that jobs in the current year pay more than jobs in lockdown (2020) although this can be contributed to many factors such as inflation and the inability for people to leave their houses during that year.

The difference is significant.

**Questions – Jake**

My first question (Question 7) was “What are the job titles for the lowest ten paying jobs (in ascending order)?”

A computer screen shot of text

AI-generated content may be incorrect.

At first it creates the job\_salary variable to find the mean of all job titles. It then moves onto creating a low\_title variable, this is the variable that will be output at the end as it will be the variable that contains the list of the lowest paying job titles on average. It sorts the job\_salary we just made into ascending order, and only selects the first ten as that is what we have set “head”to. Then we format the dataframe so that we have an easier to read output at the end.

This is the output :

A screenshot of a computer screen

AI-generated content may be incorrect.

As we can see an analytics analyst is on paper the lowest paying job you can have followed by ai software development engineer and so forth.

My next question was, what is the main trend between experience level and salary?

A screen shot of a computer code

AI-generated content may be incorrect.

Similarly to my previous question, I started by creating a variable for the output, avg\_salary\_exp. I grouped both experience level and salary in usd and then obtained a mean salary of each experience level. It then prints this so that the graphical output can take place. (Bar chart). I also added a grid to help understand the values a bit better as on a blank graph it can be misleading and accurate values may be harder to withdraw.

The output for the values is as follows:

A screen shot of a computer

AI-generated content may be incorrect.

This may be hard to read and compare so the added benefit of the graph helps massively to visualise it.

A graph of a salary

AI-generated content may be incorrect.

As we can see the grid lines help a lot when it comes to seeing the values. For example with the lines we can see that Entry levels jobs are just above 100,000 whereas if the lines weren’t there then we may say below.

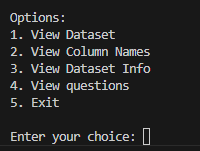
**Questions - Andrei**

**Questions - Prajesh**

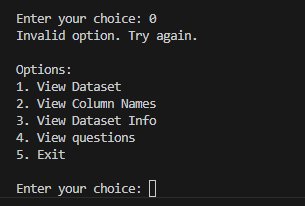
**Questions - Faris**

**Testing**

Performing rigorous testing to make sure the program works as intended is required. In order to make sure the code doesn’t stop at unexpected points many pieces of validation have been included starting with in the main menu:



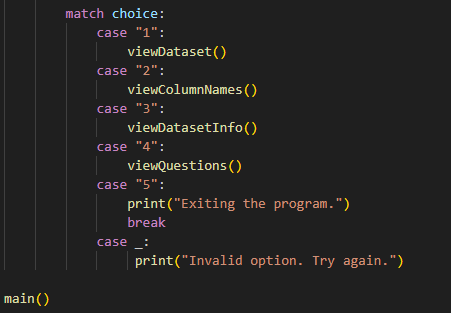
The intended input for the main menu is numbers from 1-5 but what happens when a number above 5 or below 1 is input?



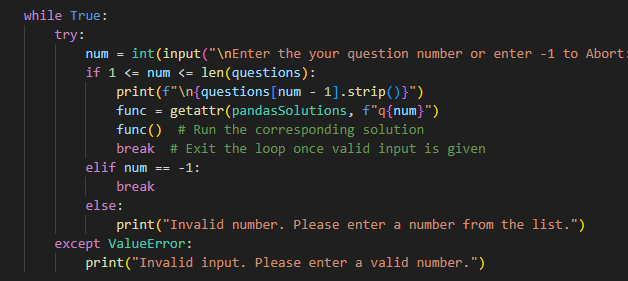
An error message is displayed and the code loops to ask the user for a valid response making it clear what the user has input incorrectly.

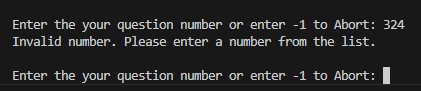


The result is identical when letters are input

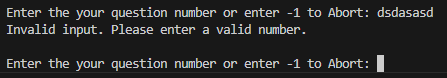


The code uses a case structure to provide responses for the menu options 1-5 and gives every other response a default error message accounting for everything

The second thing to conduct testing and validation on is the question number input which uses a try catch structure to account for its unique retrieval of the questions accounting for both invalid numbers and letter responses.



If a number entered isn’t present within the questions file an error message is displayed and user is prompted to provide another input.

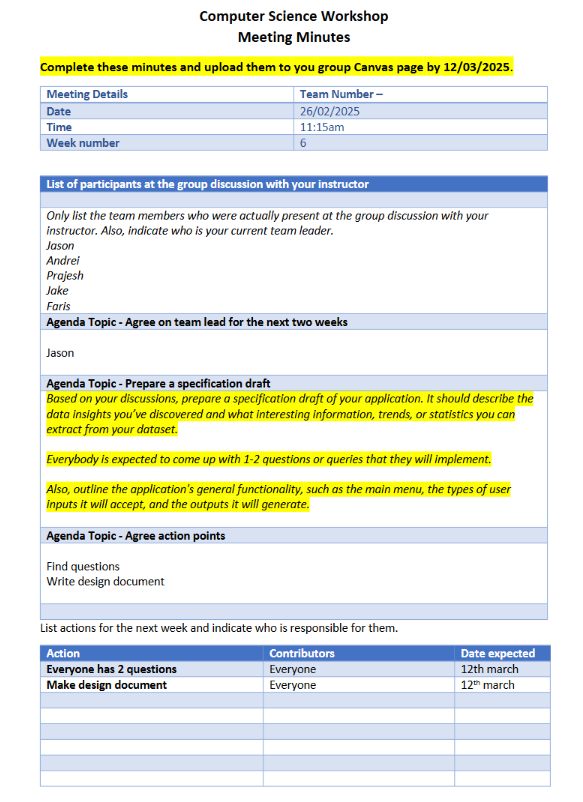


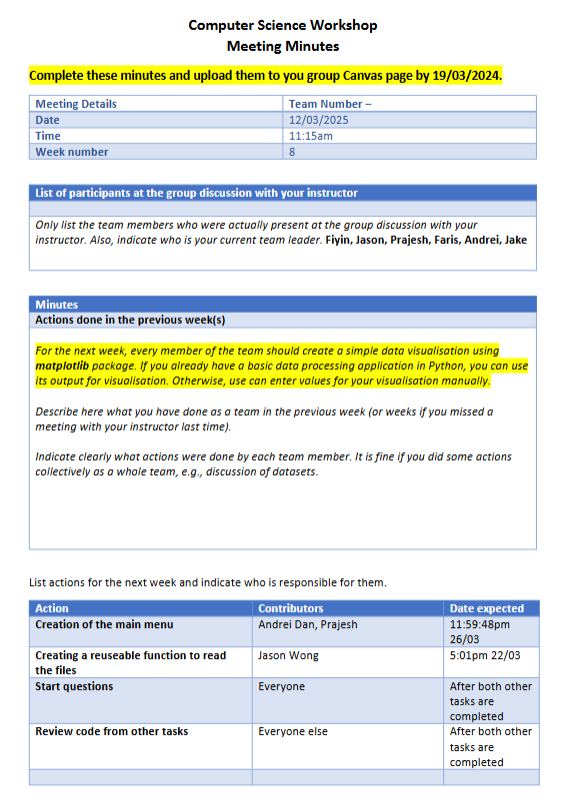
For non-number responses a ValueError is used to detect these allowing a separate error message to be displayed apart from the one used for a number relating to a question that doesn’t exist.

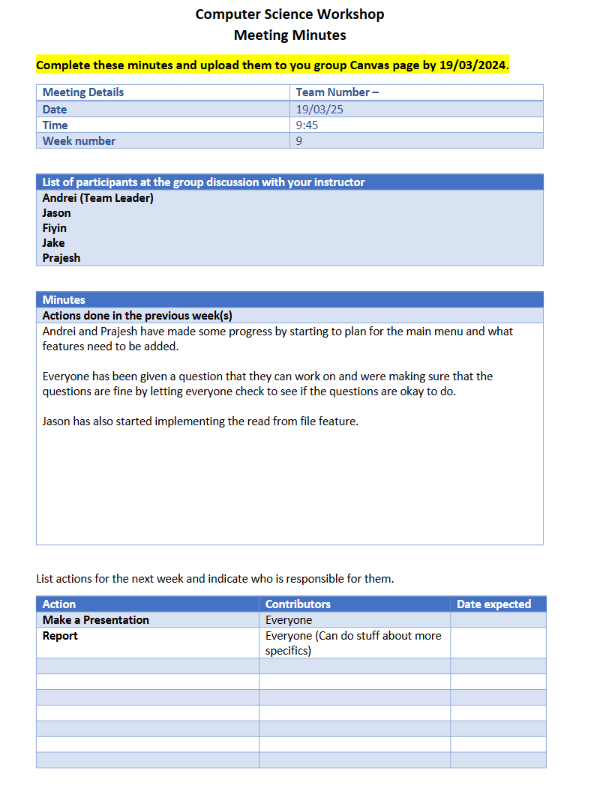
**Meeting Minutes**

**A screenshot of a computer

AI-generated content may be incorrect.**A screenshot of a computer screen

AI-generated content may be incorrect.



 A screenshot of a computer screen

AI-generated content may be incorrect. A screenshot of a meeting minutes

AI-generated content may be incorrect.

**Group Evaluation**

Challenges: merging conflicts creating potential losses in progress and long periods of checking files for incorrect changes, finding periods where the whole group can communicate progress on the project, dataset chosen fields sometimes being dominated by one value rigging the results.

Benefits: Introduced the usage of git and github to the members of a group and helped show how teamwork in the field