Creating an Income Calculation Model – Group 8

Introduction

This task was approached following the learning outcomes that were provided to us at the top of the Jupyter Notebook document. As well as looking to create a calculation that transformed gross annual income into monthly net income and vice versa, we sought to understand this system as comprehensively as we could in the interests of completing task 2 and presenting on the subject.

Methodology

After inspecting the data and ruling out the need for any transformations, we deemed that creating a function - 'applying_tax' - that calculated net monthly salaries from gross annual salaries was an appropriate first step. For simplicity, these values for Scotland and the rest of the UK were stored in two separate lists. From here, we realised it would be simpler if data were stored separately in general for the two locations. Hence, two separate data frames were created: one for Scotland and one for the rest of the UK.

In the interests of conciseness and readability, we thought that storing the band requirements in lists and calling upon the appropriate value when required – rather than creating 'if' statements for each band – would be appropriate. The only possible way this could lead to confusion is that to successfully apply this into the function it required minimum and maximum incomes to meet the criteria. This is why, as per line 17 of the third code chunk, there is an 'infinite maximum income' for the highest tax bracket.

From here, we were able to apply for the correct rates of income tax and National Insurance for both data frames using our function, which matched each income with the correct band of tax and used the 'deductions' loop to calculate the net annual income. For the rest of the UK, there are less tax brackets, therefore the loop for applying income tax was amended to adhere to this. As National Insurance is the same for the rest of the UK as it is for Scotland, the same code used previously was repeated for this step.

With this understanding of how to obtain the net monthly salaries from the annual gross income, through applying eligible tax rates and ensuring the tax band logic operated as indented; to apply the correct tax to the correct portion of income, we could tackle the problem of obtaining the annual gross income from the net monthly salary.

The function that we have created to calculate the annual gross income is named 'calculating_gross_annual_income'. This function is based primarily off the logic of the pre-discussed function that operates in the opposite direction. We needed to use the tax band system to understand what level of taxes have been taken off, however we did not have the annual gross salary to apply these tax rates to as this is the value we are attempting to obtain. Therefore, the only value we could put through these tax bands was the annual net salary (the

salary after taxes have been applied). To differentiate with the previous function, rather than the running total of deductions being used to subtract at the end, this value utilises addition.

Once the net annual salary has been passed through the eligible tax bands, this offers an estimation of what the annual gross salary would be, rather than a precise value. This is where our helper functions are essential.

From the understanding of our 'applying_tax' function, which takes in a data frame and splits the data to apply the eligible tax bands to calculate the net monthly salary, we created helper functions to do similar. The helper functions, 'applying_scottish_tax' and 'applying_rou_tax' take in a single value for gross annual salary to calculate its corresponding net monthly salary with the specific regionalised tax bands.

Returning to the 'calculating_gross_annual_income' in which we have an estimated value for the annual gross salary, this value can then be passed into the corresponding helper function. This will then calculate what the net monthly salary would be for this estimated annual gross salary. From the data we know what the net monthly salary should be, therefore we can take both net monthly salaries and compare them to understand the difference between them. From this monthly differential we can multiply it by twelve to find the annual difference between them and add it to the estimated annual gross salary we have calculated. This process moves the estimated value closer to what the precise annual gross salary should be.

A single iteration of calculating the difference will not then give the precise annual gross salary. It may need to be completed several times and possibly more for certain data entries than others. Therefore, we have used a conditional loop to repeat this process until the estimated gross annual salary becomes the true value for the annual gross salary for the corresponding net monthly salary that was given to us. This precise annual gross income can then be appended onto a list to be stored as the entire process repeats itself for the next net monthly income.

For visual clarity, we have also written into the function the capability to output the created data frames, storing the annual gross salary with its monthly net income, into an external excel file, separating the locations into two separate sheets.

Discussion

We feel that this approach and its subsequent code are strong as they cover every base that they need to without being unnecessarily complicated and unreadable. The aims of the task are also achieved through this, as we believe that given the data we were provided with, we now have the most comprehensive inference of the UK system that we could have obtained. Furthermore, we also believe that the use of helper functions and the simplification of the 'applying tax' function through the eradication of unnecessary 'if' statements are original ideas that bolster the analysis. We will be both surprised and impressed if other analysis of the data takes this initiative.

Despite this, there are some key limitations that we need to be aware of and potential threats to the integrity of our conclusions.

The data does not provide us with key contextual information such as employment status, council tax band and student loan repayments. These are key contributors that directly influence net income. This means that the net income that we have calculated may not actually be reflective of one's financial position. This negatively impacts our ability to draw any meaningful conclusions from the analysis, despite the comprehensive process. The data that we were provided with essentially gives anecdotal evidence; but we need the full picture to add the essential context and meaning behind it.

If we were given more data to account for this, we could incorporate the missing values into our analysis. If we were able to account for essential expenses such as council tax and rent/mortgage, our analysis would better reflect the financial position and quality of life of the people that the data represents. This would of course require slightly more time to incorporate into the functions, but we believe that this will bring a more realistic and personable insight to the analysis, ultimately reflecting real life more accurately than the anecdotal nature of the present study's results.

This does not discredit the present study, as tax as a standalone element is very important to consider. However, considering further deductions alongside this is crucial, hence obtaining analysis of both is vital.

To conclude, this analysis allowed us to obtain inferences about how Scotland and the rest of the UK are taxed regarding their income as much as the data set allowed. Due to the lack of consideration for other essential salary deductions such as council tax, rent/mortgages and bills, it would be naive to deem these results conclusive. Rather, we must acknowledge that this study is a solid foundation of understanding salary deductions of one key dimension (tax) but to understand this fully we need to account for the absent variables in the data set in future research.