**Income statement preparation using Python**

In this post, we will prepare the income statement of a company from the trial balance using python. We will further show the change (in percentage and amount) when compared to the prior year.

**Trial balance and income statement**

The trial balance is a report that lists the balances of all general ledger accounts of a company at a certain point in time. The accounts reflected on a trial balance are related to the five major accounting types (assets, liabilities, equity, revenues, and expenses). The income statement is a financial statement that shows the income and expenditure of a company for a given period.

To prepare the income statement from the trial balance, we will only be needing the revenue and expenses account type. The revenue accounts include all the money a company earned for services rendered during the accounting period even if payments have not been received yet. The expense account on the other hand includes all the expenses incurred by a company during the period even if payments have not been made yet. The difference between the total revenue and total expenses during the period is the net income or loss during the period.

**Preparing the income statement with python**

For us to prepare the income statement with python, below are the steps we are going to follow:

* Get the trial balance for the period
* Obtain all the accounts relating to revenue
* Obtain all the accounts relating to expenses
* Get the total revenue and expenses for the period
* Get the net income for the period
* Merge the revenue, expense, and net income into a single data frame
* Show the change between the prior year and the current year
* Save the income statement in a formatted Excel using Pandas *ExcelWriter* function

To start, let’s import the Pandas library and the trial balance into a Pandas Data Frame object.

<code for import>

<picture of output>

Looking at the trial balance, we have a column called account type. As explained earlier, this account type is grouped into the five major accounting types. This can be seen when we look at the unique values in this column using the pandas *unique()* function.

<code for unique>

<picture of output>

The next step is to obtain all the accounts relating to revenue and expenses account type. They will be saved in different data frames for us to obtain the totals easily. The revenue/expense data frame will include all accounts relating to revenue/expenses.

<code for revenue/expense dataframe>

<picture of output>

Let’s start with the revenue account type.

<code for revenue.head()>

<picture of output>

First, as we can see above, we will not be needing the Account Code column for the income statement. Let’s drop it. Next, we will group our data frame by the account class and sort the values using the closing balance column in descending order. Due to the fact that our revenue is usually a credit account, we have negative balances. We want to show the revenue figures as a positive value so we will multiply the figures by minus 1 (-1). We will then get the total revenue by getting the sum of the ‘Opening balance’ (which is the prior year) and the ‘Closing balance’ (which is the current year) and append this row called ‘Total Revenue’ to the revenue data frame. I added an empty row in this data frame which will be help separate the revenue from the expense when saving this as an excel file.

<code for revenue>

<picture of output>

We will also apply the same steps to the expense data frame to get the total expense.

<code for expense>

<picture of output>

We have gotten the total revenue and total expense so far. For us to get the net income/loss, we will need the total revenue and total expense we have derived earlier on. Using the *iloc* function, we will extract the lines of total revenue and total expense into a new data frame object called ‘net income’. In a similar fashion in which we got the total revenue and total expenses row, we will get the net income for the period by summing the total revenue and total expense and appending it to the ‘net income’ data frame object.

<code for net income>

<picture of output>

**Bringing it all together**

We currently have the revenue, expenses, and net income in separate data frame objects. For us to bring them into one, we simply have to append them to one another into a single data frame. After merging them into one data frame, we can then get the change between the current year and the prior year.

<code for change>

<picture of output>

**Exporting it to Excel**

After this is done, we can then write the combined data frame to an excel file. We can do this using the pandas *to\_excel()* function. However, because we want to be able to format the excel file before saving it, we will be using the *ExcelWriter* module. We will not go into the details of using the ExcelWriter as this is not within the scope of this post.

<code for revenue>

<picture of output>

The last two lines are used to automatically start the excel file after saving.

picture of excel output>

You can break down the items of the income statement further (for example, you may want to show the gross profit) by grouping the items based on the account class and getting the total for each account class before merging them.

In a future post, we will look at preparing a monthly income statement with python using the journal entries.

Thanks for reading.