eSTEeM:  
Measuring qualification effects framework

Table of Contents

[Introduction 1](#_Toc467054516)

[Pre-requisites 1](#_Toc467054517)

[Using the framework 2](#_Toc467054518)

[Review use of external resources 2](#_Toc467054519)

[Use of external resources - a caveat 5](#_Toc467054520)

[Use of external resources - separate module discussion 6](#_Toc467054521)

[Tracking posts during a course 8](#_Toc467054522)

[Using external resources for visualisation 14](#_Toc467054523)

[Draw a line chart tracking posts 14](#_Toc467054524)

[Draw a stacked bar chart for classified posts 19](#_Toc467054525)

[Draw a stacked percentage chart of classified posts 23](#_Toc467054526)

# Introduction

As part of this eSTEeM project we have developed a framework to support our analysis and provide a model for similar future projects. This cookbook expands upon the use of that framework.

The cookbook is divided into two sections. The first section shows you some of the framework’s functionality. The second section demonstrates how to use external resources to visualise some of the framework’s data outputs.

You can read more background information about the framework in the accompanying Report document. The report includes more detail on the workflows and their output reports and data files.

We hope you find the recipes in this cookbook useful and inspire you to consider other ways you can use the framework.

## Pre-requisites

To follow the recipes in this cookbook you will need to install the esteem-mqe framework and forum data. You will also need to ensure that your system has Python 3 installed. The Tracking posts during a course recipe also needs the Python nltk and matplotlib packages. For more information see docs/quickstart and docs/installation.

# Using the framework

## Review use of external resources

This recipe takes you from the raw forum XML data in the forums folder to a readable filtered posts report in the reports folder. The report helps you assess students’ engagement beyond immediate course material based on whether the post contains a reference to an external resource.

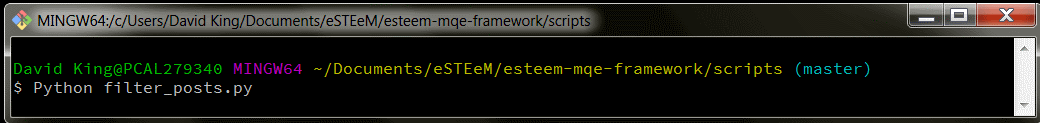


Figure 1: Run filter\_posts.py

Open a command prompt in the scripts folder and run the filter\_posts.py script. See Figure 1: Run filter\_posts.py on page 2.

Note. As currently written the framework is a series of scripts run from the command line. An improvement in this means of running the scripts is one of the technical enhancements documented in script/TODO.

By default, the script processes all forums in the forums folder, and writes the filtered reports to the reports folder. The forum names may be different in the final delivered sample set from those shown in these screenshots. The script creates the reports folder, if it does not already exist.

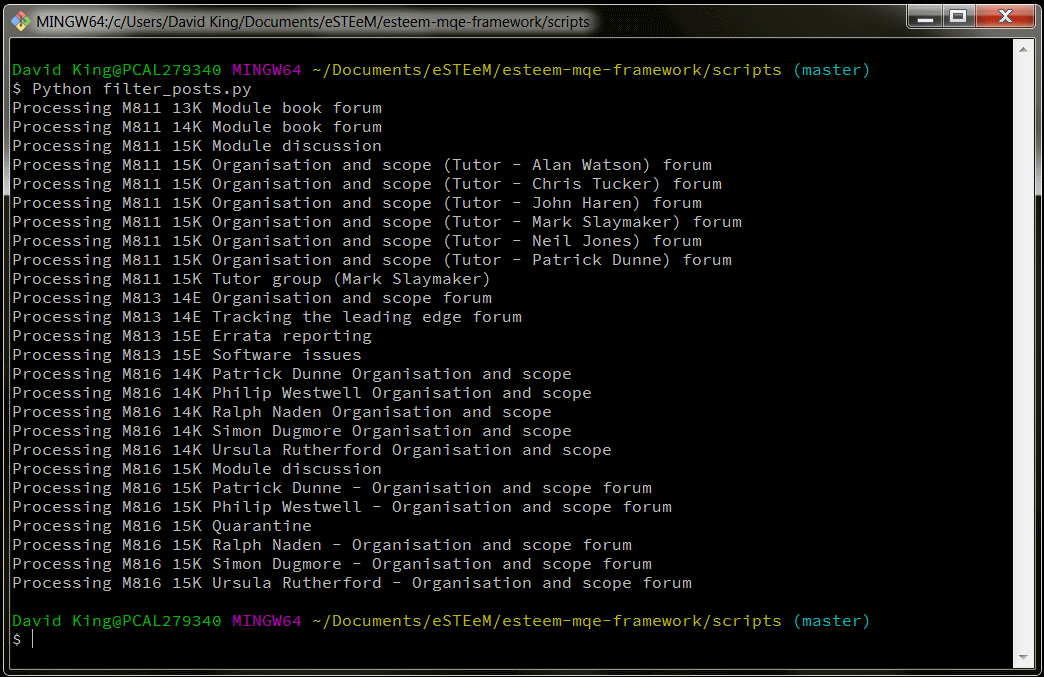


Figure 2: Messages from filter\_posts.py

The script takes only a couple of seconds to run. The script confirms which files it processes, as shown in Figure 2: Messages from filter\_posts.py on page 2.

Running filter\_posts.py script, populates the forums folder with the filtered posts reports and an index file, as shown in Figure 3: Listing of reports folder on page 3.

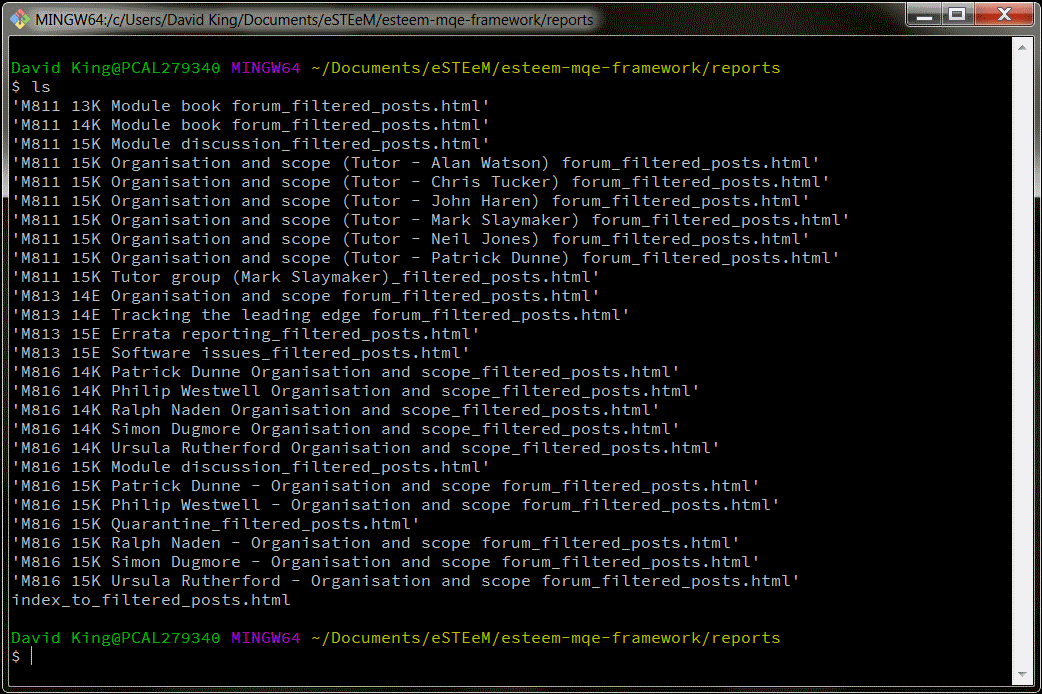


Figure 3: Listing of reports folder

All reports are html files, and hence are read by opening in a web browser. The screenshot in Figure 4: Index to filtered posts reports on page 4 shows the top part of the index page created by filter\_posts.py. Simple metrics about each forum is presented, showing the number and percentage of posts with and without external references. The forum names themselves are links to each forums’ own report page.

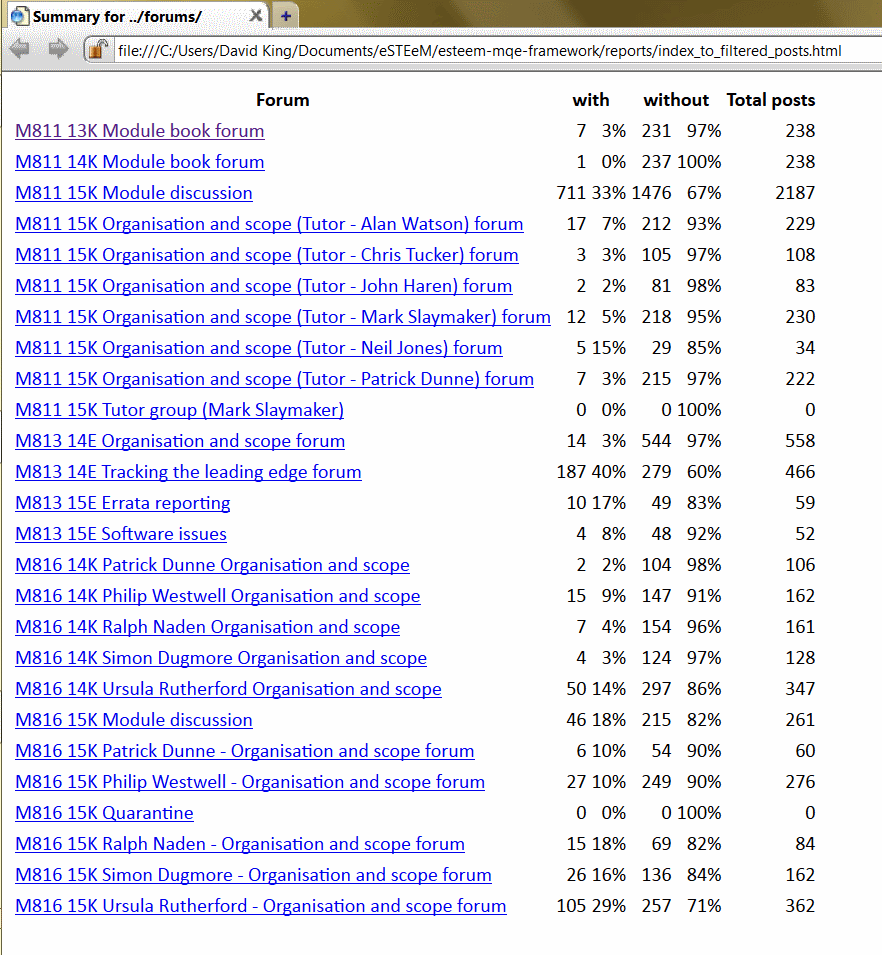


Figure 4: Index to filtered posts reports

Clicking on the link to M816 15K Ralph Naden – Organization and scope forum, takes you to its report. See Figure 5: Filtered post report - relevant example on page 5. The screenshot of the report demonstrates the original motivation behind developing this framework: in a matter of seconds, by running one script, thousands of relatively inaccessible posts have rendered into one consolidated, easy to read report.

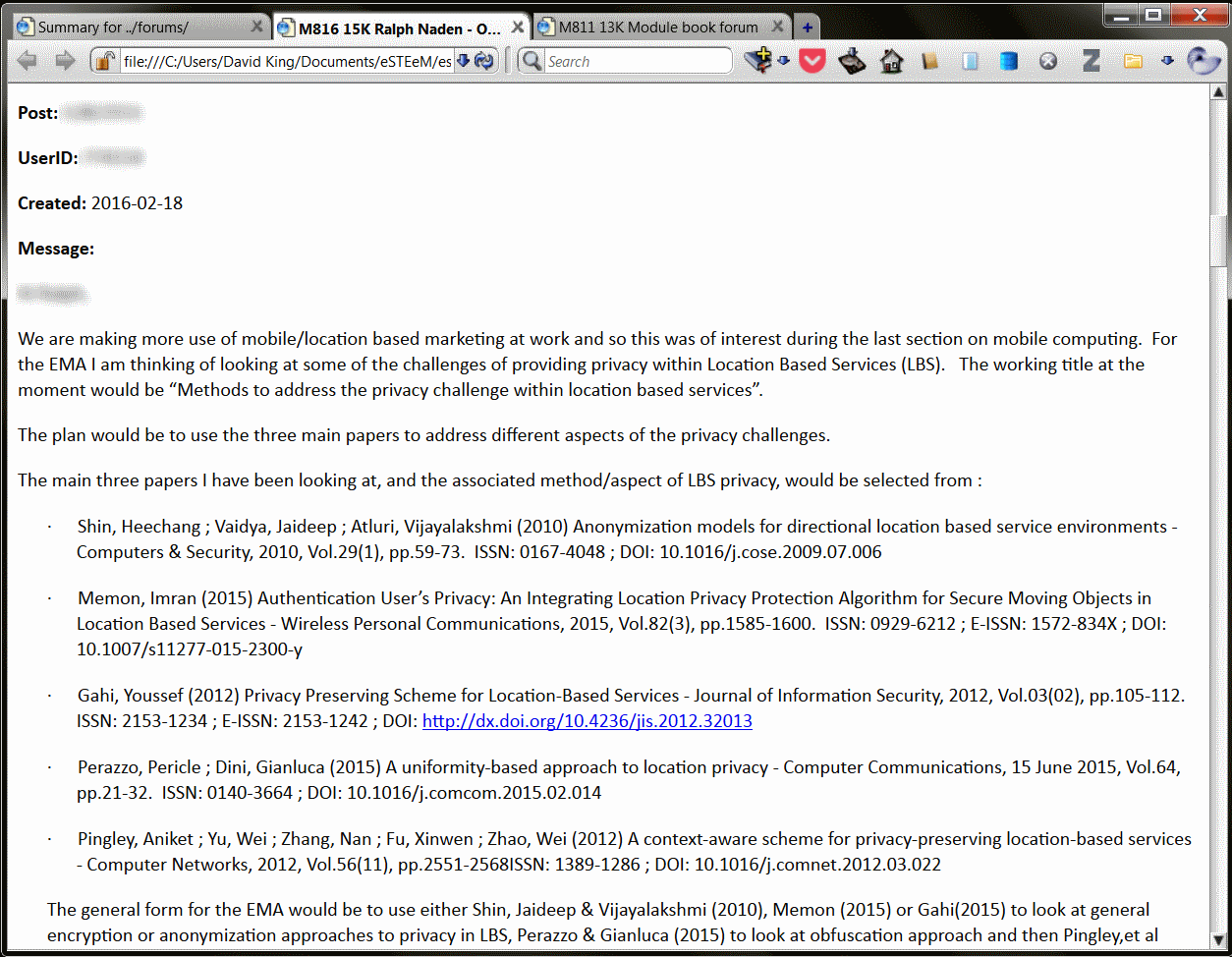


Figure 5: Filtered post report - relevant example

This screenshot in Figure 5 shows one post from the forum’s report to enable its close reading. In this case, the post’s author had drawn on several external sources, two of which are very recent. In addition, you can see that the references and citations are formatted according to OU guidelines. The framework has saved you, the academic reviewer, the time and effort necessary to get straight to the post and assess its quality.

### Use of external resources - a caveat

As implemented, the framework only looks for external references. There is no assessment of their quality. You can see the consequence of this on clicking on the link to M811 13K Module book forum to view its report.

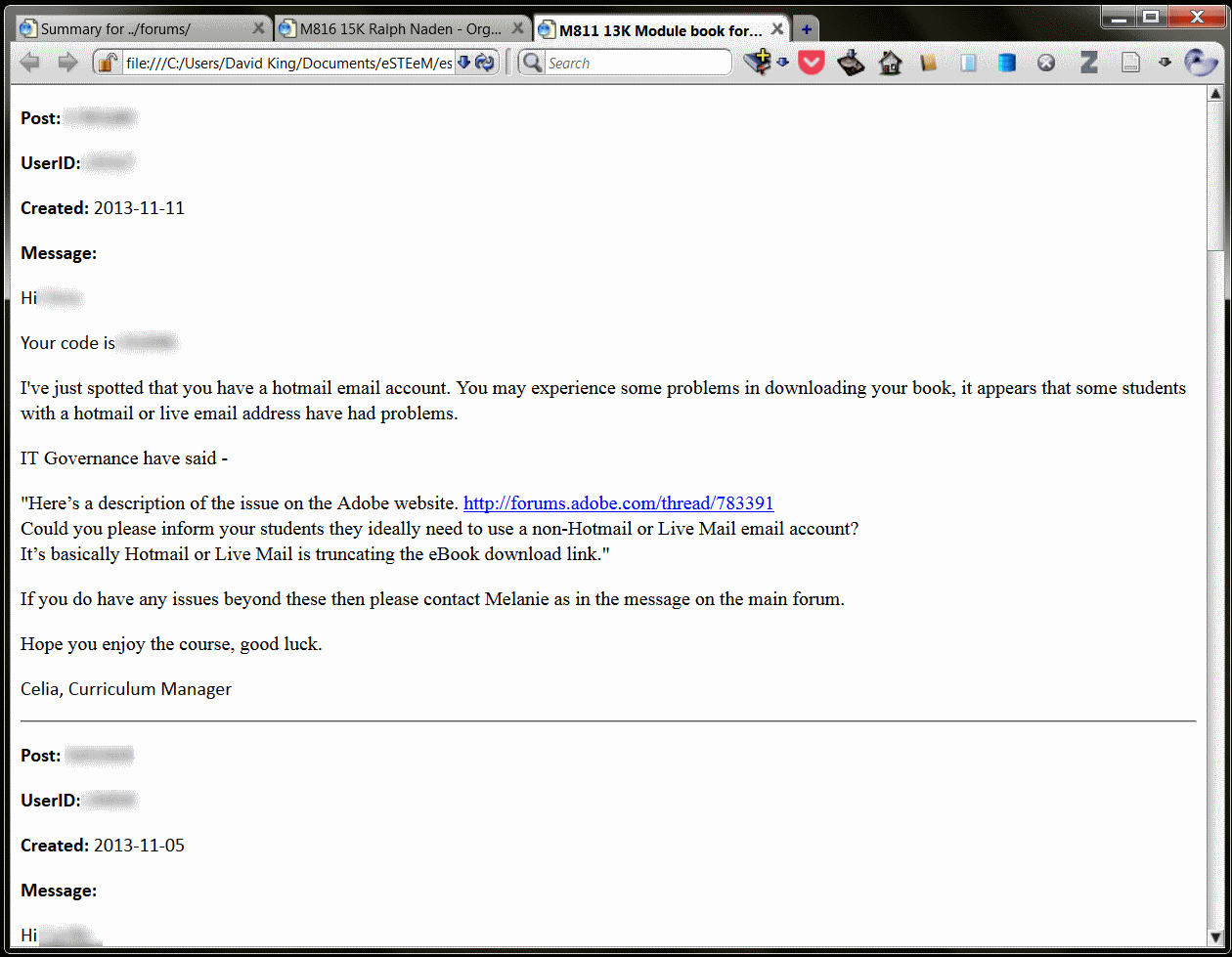


Figure 6: Filtered post - misleading example

This forum is an administrative forum, helping students use the course book, which is provided online. Hence, simply applying the framework to this forum will identify external resources, in this case a link to Adobe regarding a problem accessing the online book. It tells us nothing about the pedagogical intervention under examination in the current study.

Therefore, before using the framework it remains necessary to understand your data.

### Use of external resources - separate module discussion

The Module book forum does not contribute to our study, except to provide additional test data when developing the framework scripts. The forums that contribute most to our study, and will contribute most to other studies of pedagogical interventions, are module discussion forums.

Separating out student interaction from administration isolates the indicators we want to assess in our study. An excellent example of this can be seen in examining M816 15K Module discussion forum, the report for which is shown in Figure 7: Filtered post - a perfect example of student discussion on page 7.

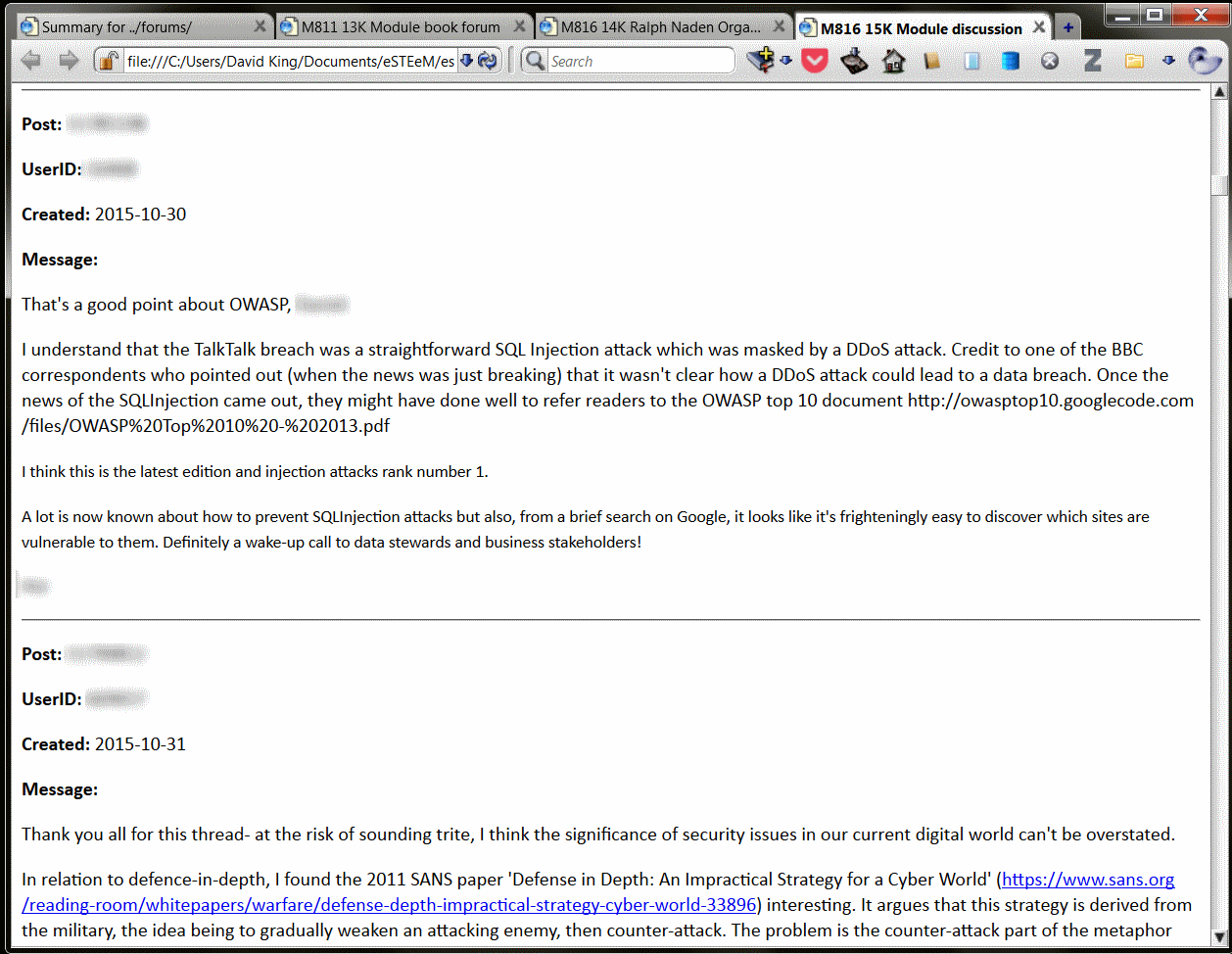


Figure 7: Filtered post - a perfect example of student discussion

Use of the framework has filtered out 46 of the 261 posts in this discussion forum. That means in assessing the impact of the teaching interventions in our study, firstly, we knew that only 18% of the posts were relevant (and what that tells us about student engagement with external resources) and secondly, that we could focus our close reading on those posts in one report without having to select them individually from the forum. This recipe helps you achieve similar benefits with your examination of teaching interventions as realised in student discussions.

## Tracking posts during a course

This second recipe looks at how to track the volume of posts during a course, comparing the total number of posts made each day with those that meet the filtering criteria. This will enable you to assess student engagement with course activities with reference to the course calendar. The sort of question this recipe can help answer is whether student engagement responds more to setting tasks or to meeting TMA deadlines, with the volume of forum posts serving as an indicator of the students’ engagement with the work in-hand.

This recipe assumes you have already completed the previous recipe Review use of external resources and run filter\_posts.py.

To run the framework scripts open a command window in the scripts folder and run the classify\_posts.py script. See Figure 8: Run classify\_posts.py on page 8.

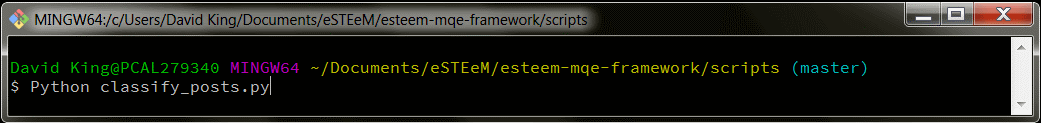


Figure 8: Run classify\_posts.py

By default, the script processes all forums in the forums folder. See Figure 9: Messages from classify\_posts.py on page 8. The script writes its output to the posts-classified folder. The script creates the posts-classified folder, if the folder does not already exist.

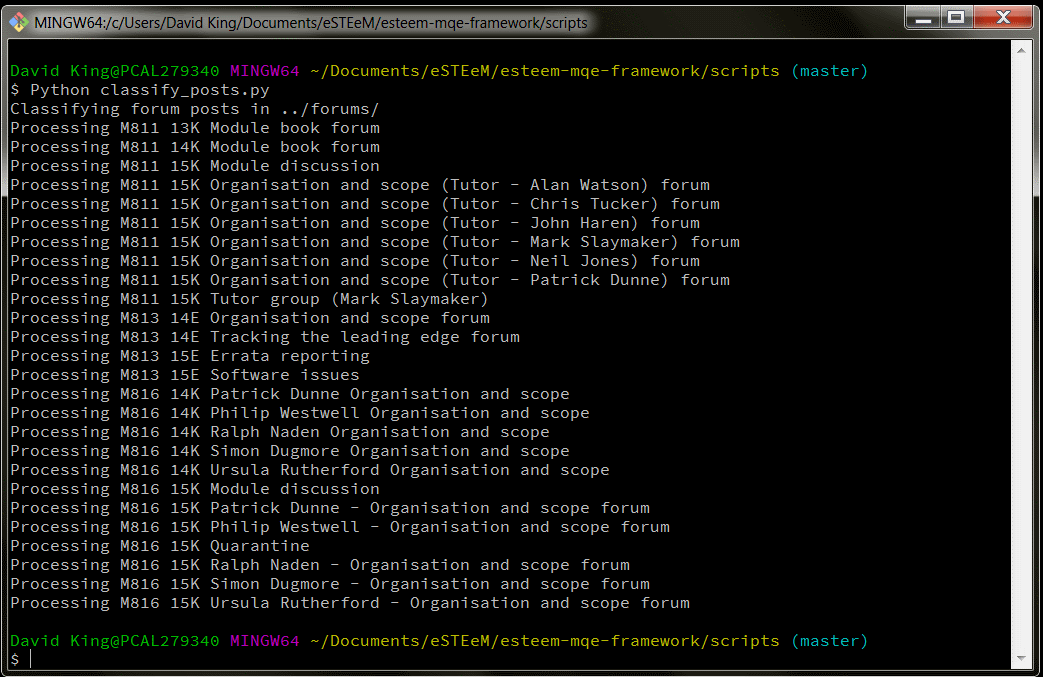


Figure 9: Messages from classify\_posts.py

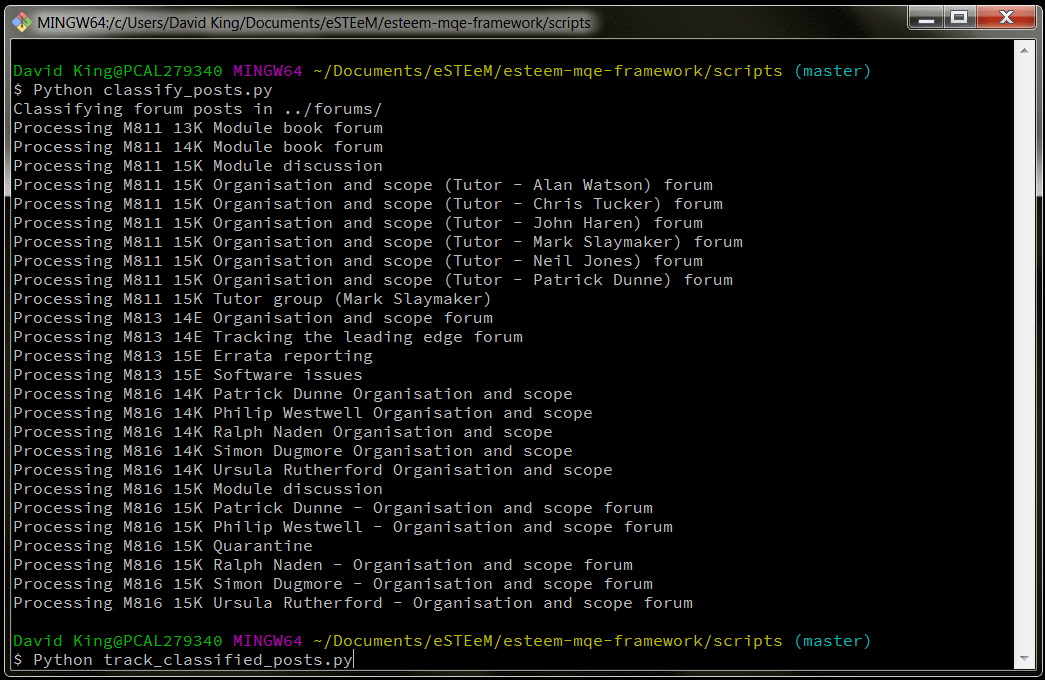


Figure 10: Run track\_classified\_posts.py

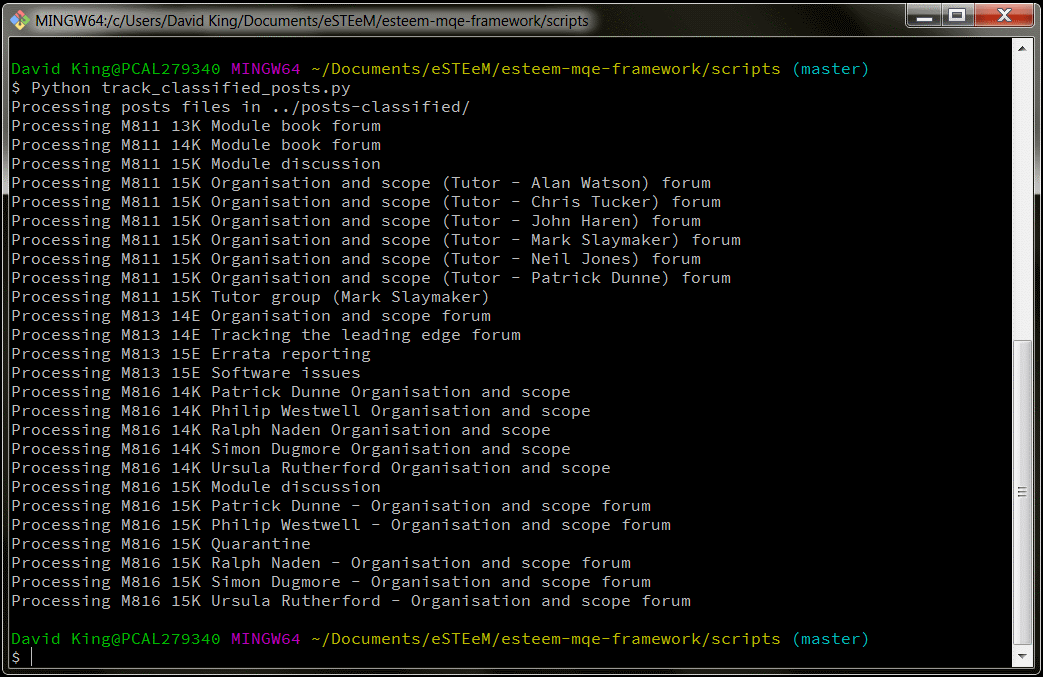


Figure 11: Messages from track\_classified\_posts.py

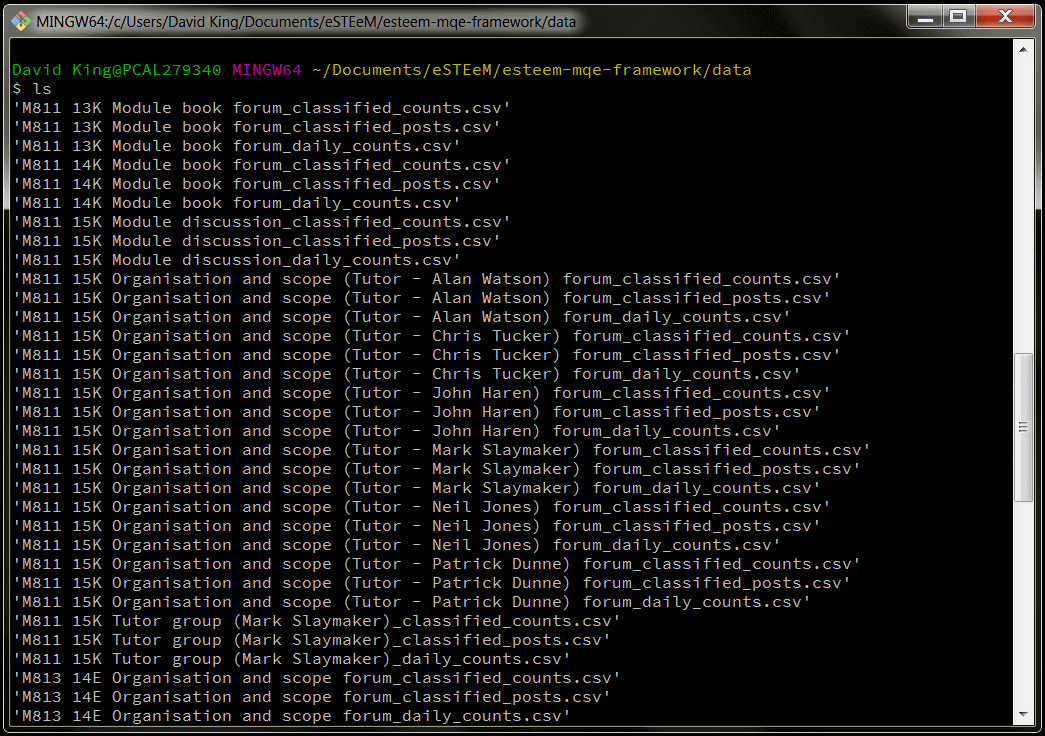


Figure 12: Listing of data files after running track\_classified\_posts.py

Having classified the forum posts, the next step is to look at them over time. To do this run track\_classified\_posts.py. See Figure 10: Run track\_classified\_posts.py on page 9.

By default, the script will process all posts in the posts-classified folder. See Figure 11: Messages from track\_classified\_posts.py on page 9. The script writes its data output to the data folder. The script creates the data folder, if the folder does not already exist.

Figure 12: Listing of data files after running track\_classified\_posts.py on page 10 shows a listing of the data folder with the data files produced by track\_classified\_post.py. There are three files for each forum. For more information on the files refer to the Data section in the framework Report.

The file *forumname*\_daily\_counts.csv tells us the total number of posts made on a given day. To find out how many of those posts meet the filter criteria run track\_filtered\_posts.py. See Figure 13: Run track\_filtered\_posts.py on page 10.

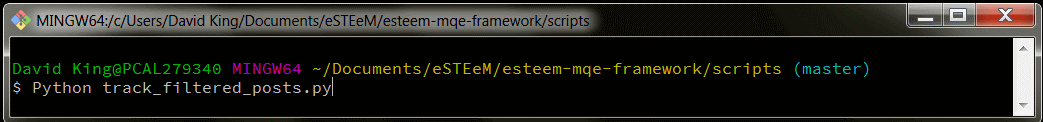


Figure 13: Run track\_filtered\_posts.py

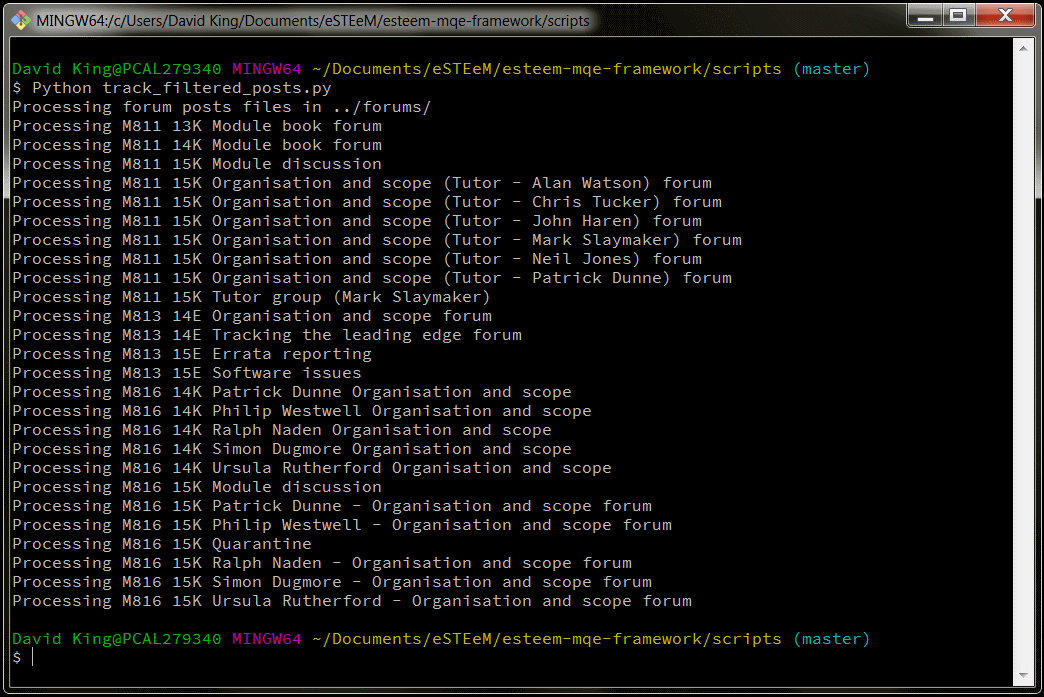


Figure 14: Messages from track\_filtered\_posts.py

The script confirms which files it processes as shown in Figure 14: Messages from track\_filtered\_posts.py on page 11. It writes additional files to the data folder, as shown in Figure 15: Listing of data files after running track\_filtered\_posts.py on page 11.

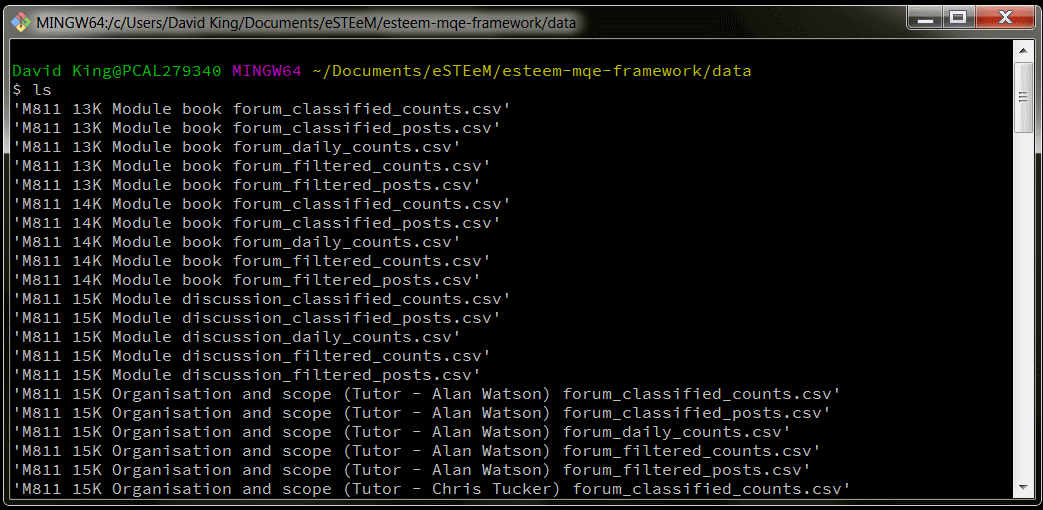


Figure 15: Listing of data files after running track\_filtered\_posts.py

There are three more data files for each forum.

The final part of this recipe is to merge the daily counts of classified posts and filtered posts, presenting the results as a line graph. To achieve this run draw\_posts.py. See Figure 16: Run draw\_posts.py on page 12.

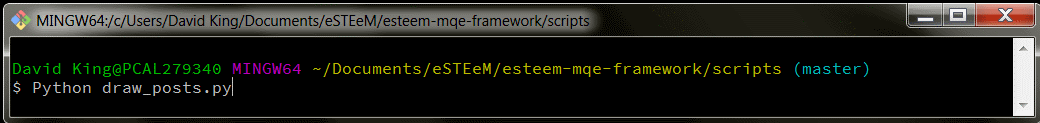


Figure 16: Run draw\_posts.py

The script confirms which files it processes as shown in Figure 17: Messages from draw\_posts.py on page 12. We process 26 forum files using the sample data, hence the warning from matplotlib about consuming too much memory. By default, matplotlib issues this warning if more than 20 files are plotted. In our case, we are drawing very simple line charts and so can disregard the message. The script writes its plot output to the images folder. The script creates the images folder, if the folder does not already exist.

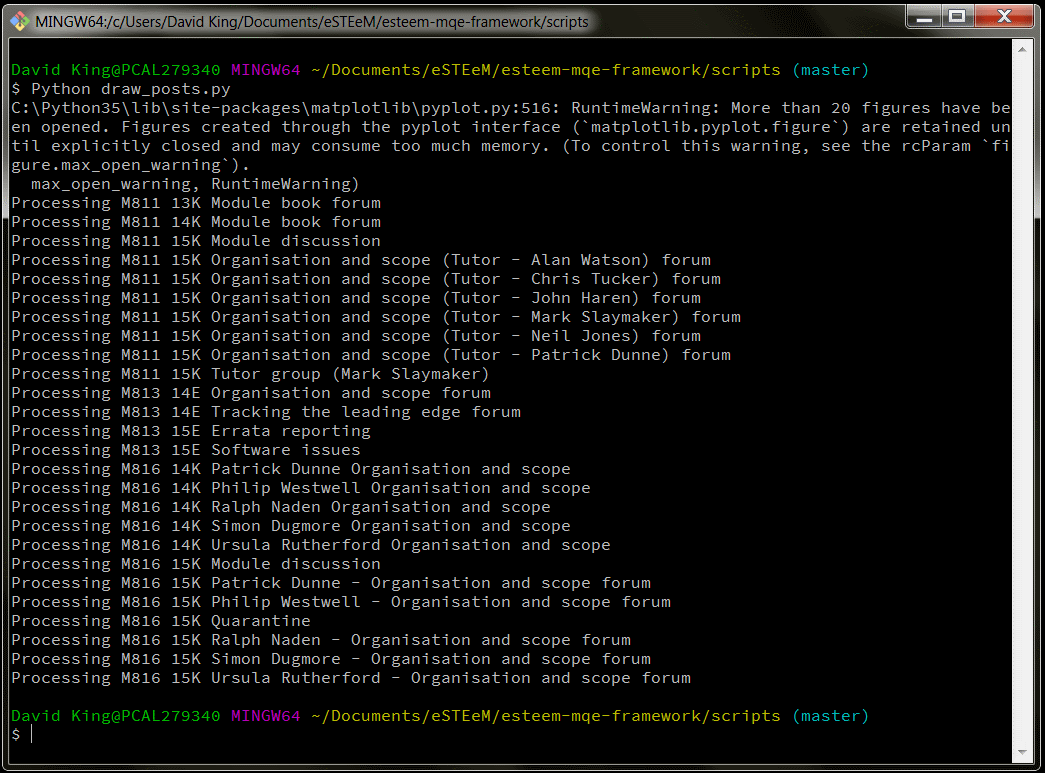
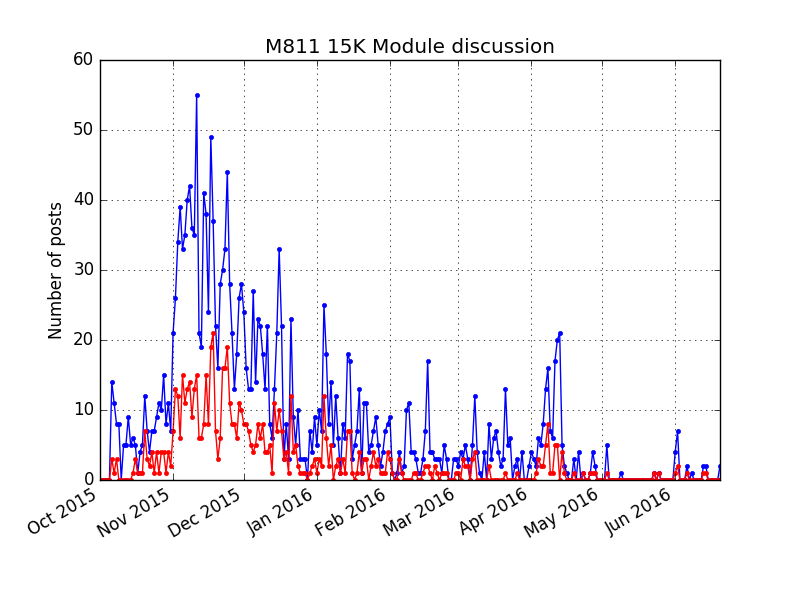
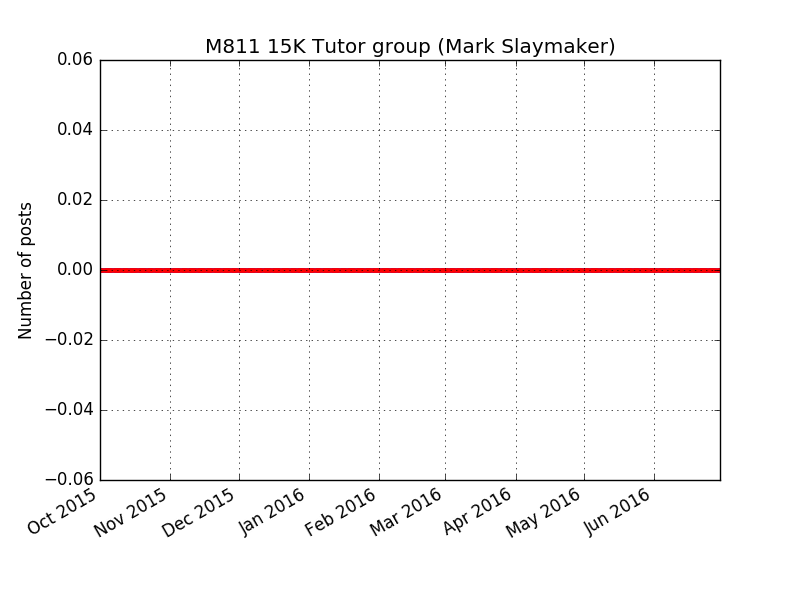


Figure 17: Messages from draw\_posts.py

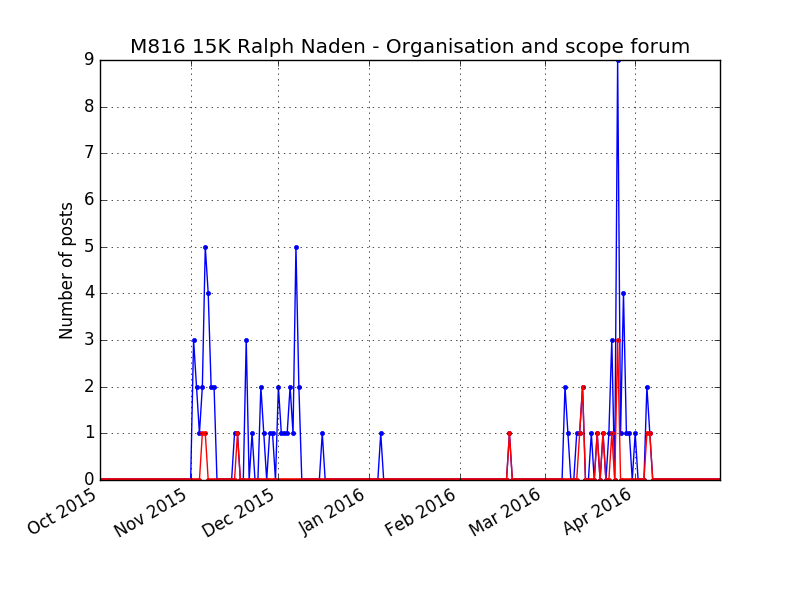
In the images folder there is now a line chart for each forum. Some samples are shown below. In all cases the blue line shows the total number of posts made that day, and the red line the number of posts made that day that meet the filter criteria. These patterns of daily posts can be related back to the course calendar.



This is a good example of discussion forum, enabling the clear tracking of forum use during a course.



This is an example of an empty forum.



This is an example of a primarily administrative forum. The posts peak in line with eTMA and similar course activities.

However, while these line charts are useful, you can achieve more if you follow the recipes in the second part of this cookbook.

# Using external resources for visualisation

The recipes in the second half of this cookbook assume you have completed the recipes in the first part. This means you have six data files for each forum in the data folder. All of these files are comma separated format files. This format can be read by many other tools. The recipes in this section use two such tools, LibreOffice Calc and Microsoft Excel to draw charts that visualise the data.

## Draw a line chart tracking posts

This recipe draws a line chart plotting the number of posts made each day during a course. This helps you assess the use of the forums, and how that relates to course activities.

This recipe uses LibreOffice Calc. First, open the file data/ M811 15K Module discussion\_daily\_counts.csv in Calc. You will be presented with an import dialog box to confirm the content of the file.

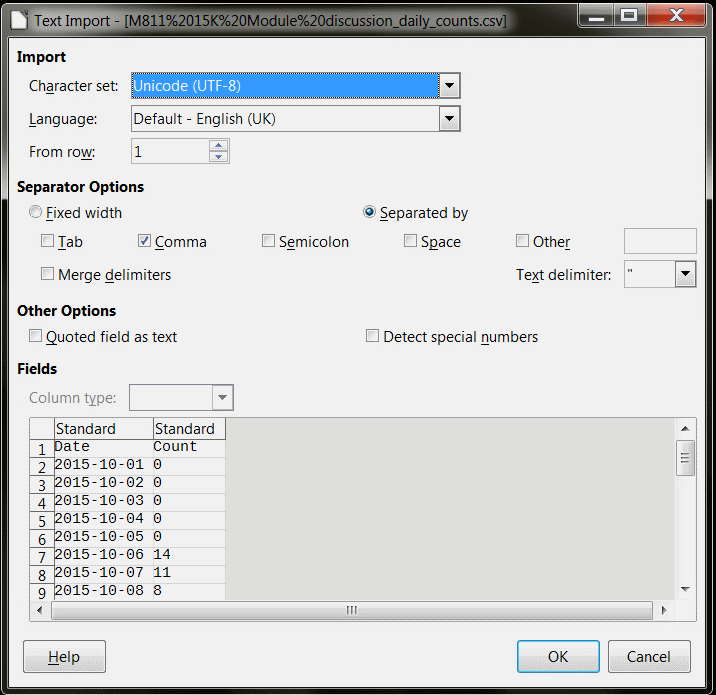


Figure 18: Import dialog in LibreOffice Calc

LibreOffice does not make assumptions about your file’s format. It attempts to identify the format details, and presents a dialog box to confirm them. Use the responses shown in Figure 18: Import dialog in LibreOffice Calc on page 14.

All esteem-mqe-framework data files follow this format.

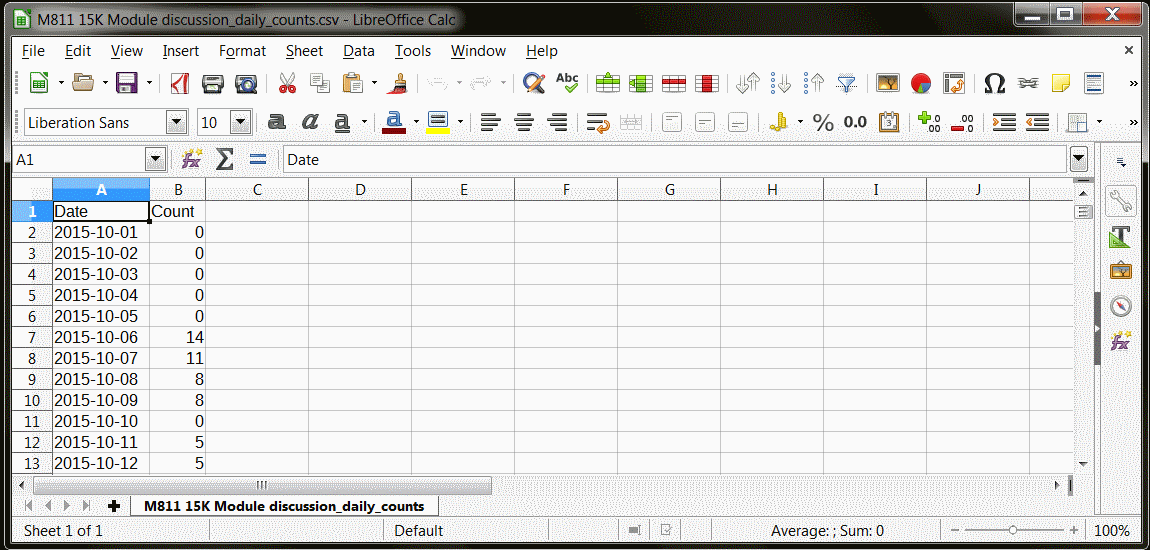


Figure 19: Daily\_counts imported into Calc

Then select the data columns. See Figure 20: Selected data columns in Calc on page 15.

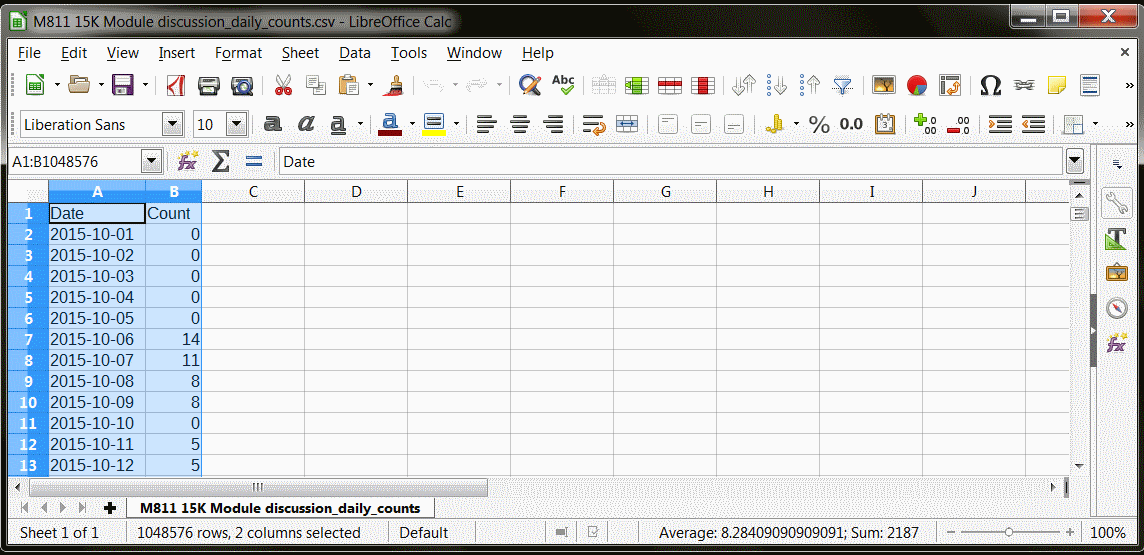


Figure 20: Selected data columns in Calc

From the menu select Insert, and Chart. See Figure 21: Insert Chart in Calc on page 16.

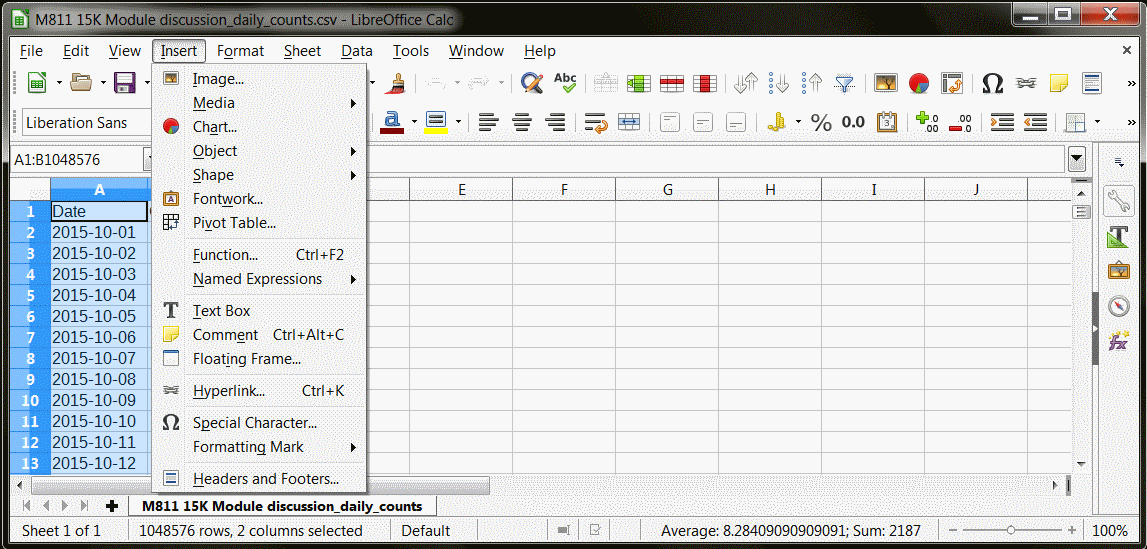


Figure 21: Insert Chart in Calc

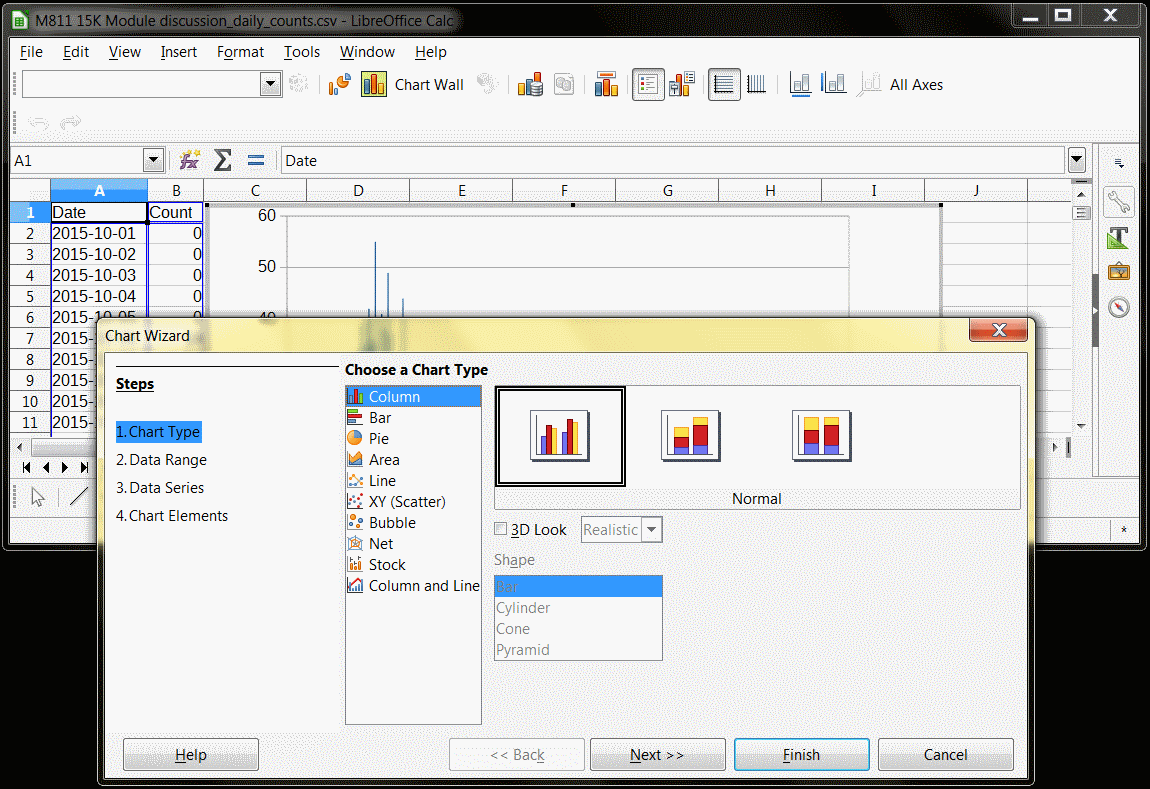


Figure 22: Chart wizard in Calc

This starts Calc’s Chart Wizard. By default, the wizard draws a simple bar chart. Select Line from the **Choose a Chart Type** options, and then Lines Only. See Figure 23: Select line chart in Calc on page 17.

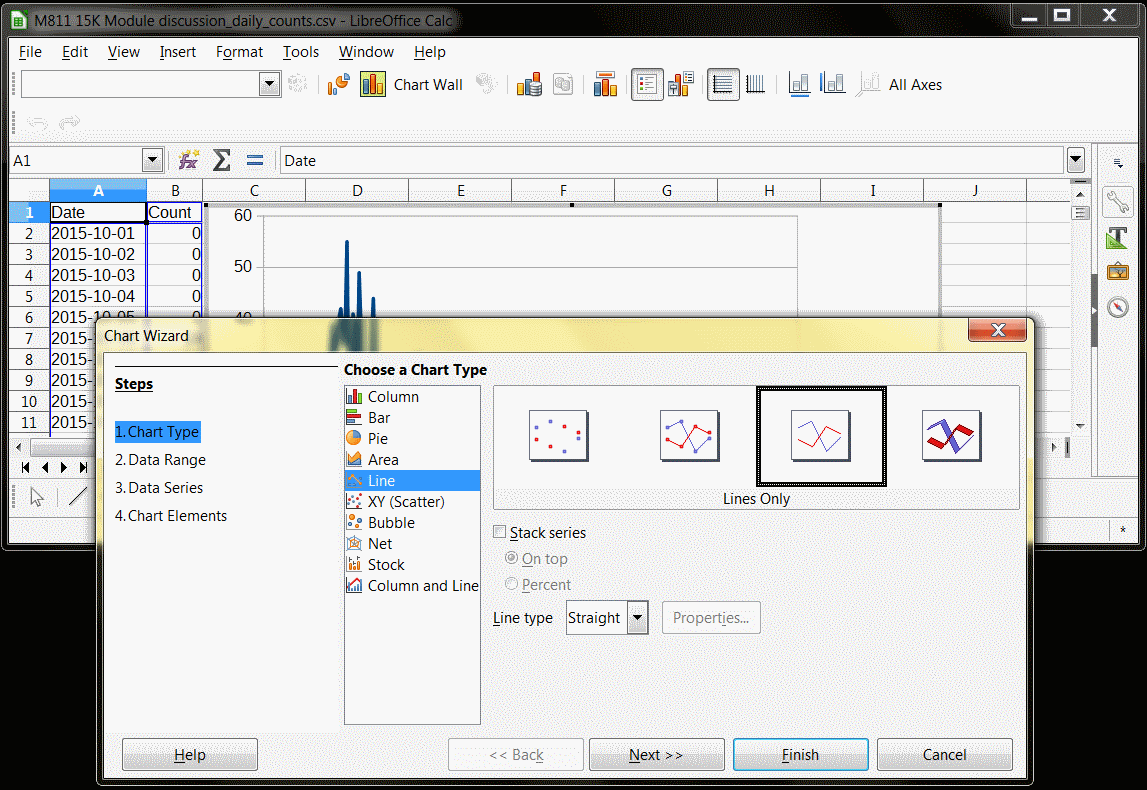


Figure 23: Select line chart in Calc

Then click Finish to produce the line chart in Figure 1: Run filter\_posts.pyFigure 24: Line chart plotted in Calc on page 18.

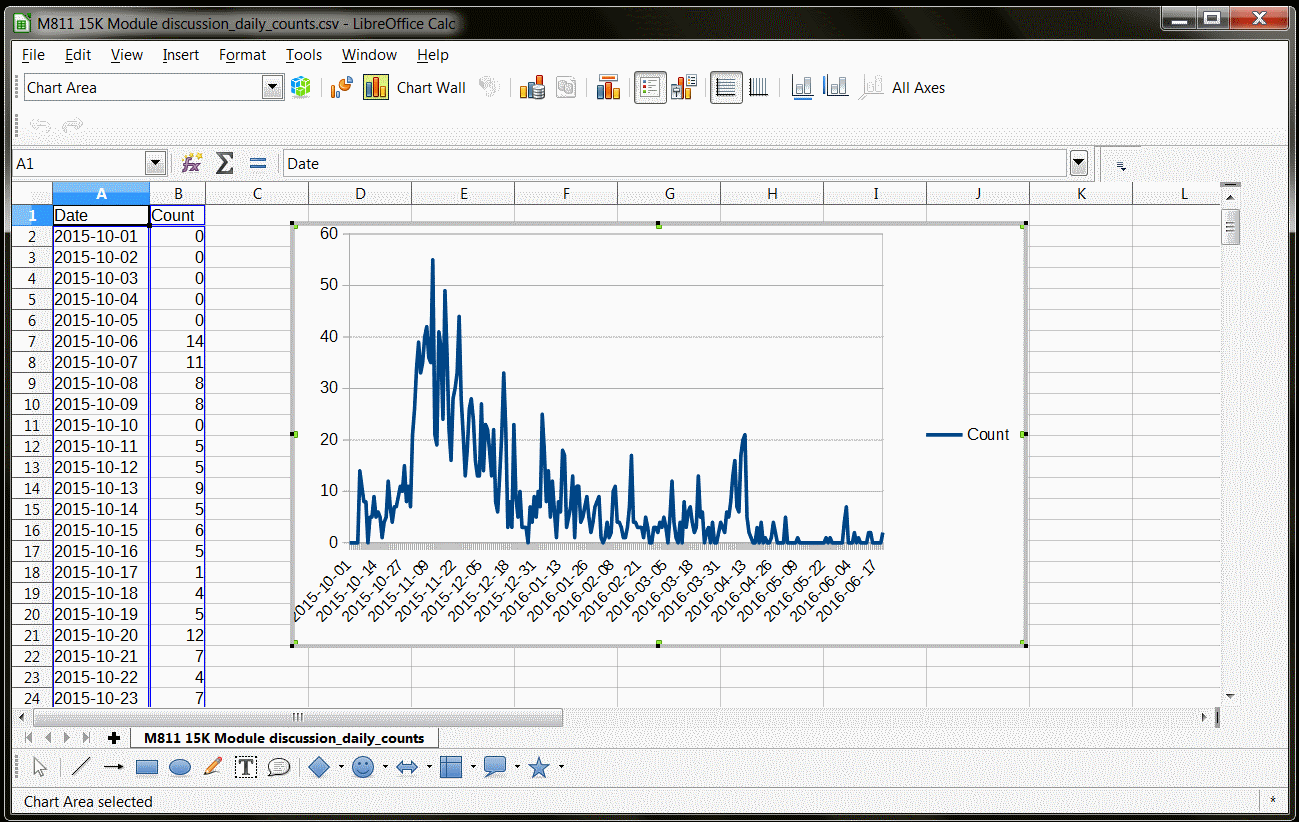


Figure 24: Line chart plotted in Calc

You can tailor the chart using the Insert and Format menus to better meet your needs. You can add a Title to the chart, adjust the X and Y axis scales and labels, change the line colour and so on. Many of these options are also available in the chart wizard used to create the chart initially. In addition, you can resize the chart, move it to its own spreadsheet in the workbook and make other such enhancements. These options and enhancements are beyond the scope of this cookbook. Please refer to LibreOffice’s help for more information on formatting charts.

To save the line chart you cannot use the original file’s comma separated file format. Instead, we recommend you use LibreOffice’s default the file formats. To save the chart only select .odc (Open Document Foundation Chart) format, or to save the whole workbook select .ods (Open Document Foundation Spreadsheet) format.

## Draw a stacked bar chart for classified posts

This recipe draws a stacked bar chart showing for each day of the course on which one or more posts are made. The chart shows the number of times a classification is used in a post. This helps you assess the content of the posts, to see if the nature of that content changes, and if either of these match the intended outcomes of the course.

This recipe uses Microsoft Excel. First, open the file data/ M816 15K Module discussion\_classified\_posts.csv in Excel. See Figure 25: Classified posts opened in Excel on page 19.

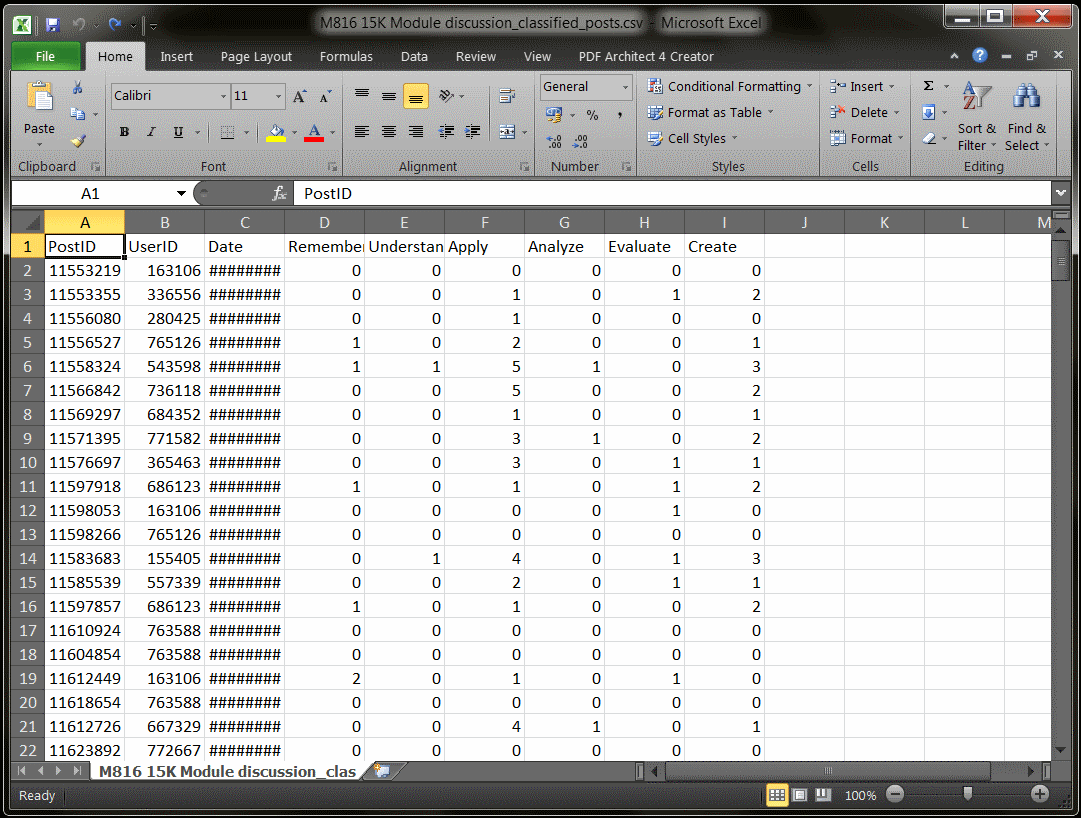


Figure 25: Classified posts opened in Excel

In contrast to opening a file in LibreOffice, you will not be presented with an import dialog box to confirm the content of the file. Normally, this will not matter because the software will make the correct assumptions about your data. However, when dealing with international students and the use of extended characters Excel might not import them correctly. The framework does handle all international characters correctly. Whether your system does or not depends on the configuration of your Windows system and is beyond the scope of this cookbook. The suggested sample data file is imported into Excel correctly, though by default the Date column is too narrow to display the values and so Excel uses hashes as a placeholder instead.

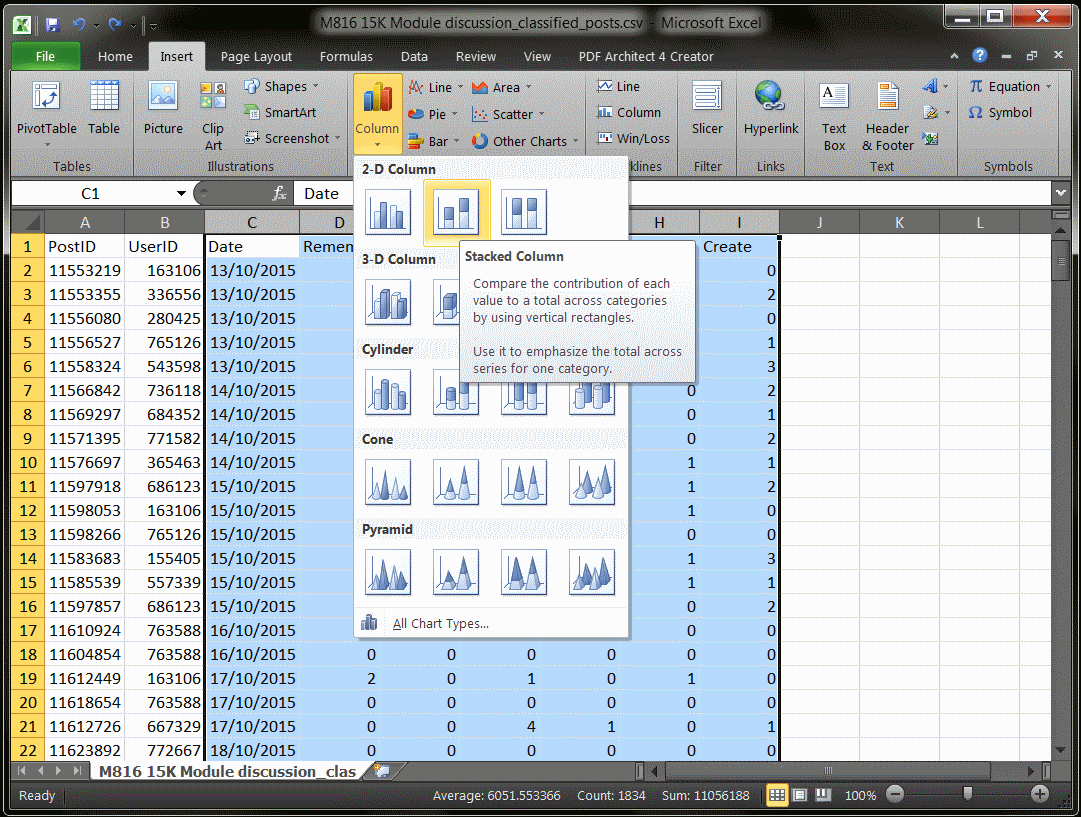


Figure 26: Select stacked column chart in Excel

Select the Date and Classification columns in the spreadsheet, columns C to I inclusive. From the Insert ribbon select the Column drop down menu and from that select the 2-D Column, Stacked Column option. See Figure 26: Select stacked column chart in Excel on page 20.

Note, to distinguish between vertical and horizontal bar charts, Microsoft uses column and bar respectively. In common parlance and as we use in this cookbook, a bar chart with vertical bars is still referred to as a bar chart.

Clicking on the selection inserts a stacked bar chart as shown in Figure 27: Stacked bar chart in Excel on page 21.

As with LibreOffice, the chart can be edited and enhanced to better meet your needs. For more information on how to do this refer to Excel’s help.

Many options are most conveniently reached by right clicking on the chart to bring up the context menu. Figure 28: Move chart in Excel on page 21 shows the context menu with Move chart selected.

A similar right click on the chart’s ‘Data Series’, ie columns, brings up the context menu shown in Figure 29: Format chart in Excel on page 22. Use this menu to remove the gaps between the chart’s columns to make the chart easier to interpret.

The final bar chart is shown in Figure 30: Formatted bar chart in Excel on page 22.

To save this chart, having created it in Excel we suggest you use Excel’s default format, .xlsx.

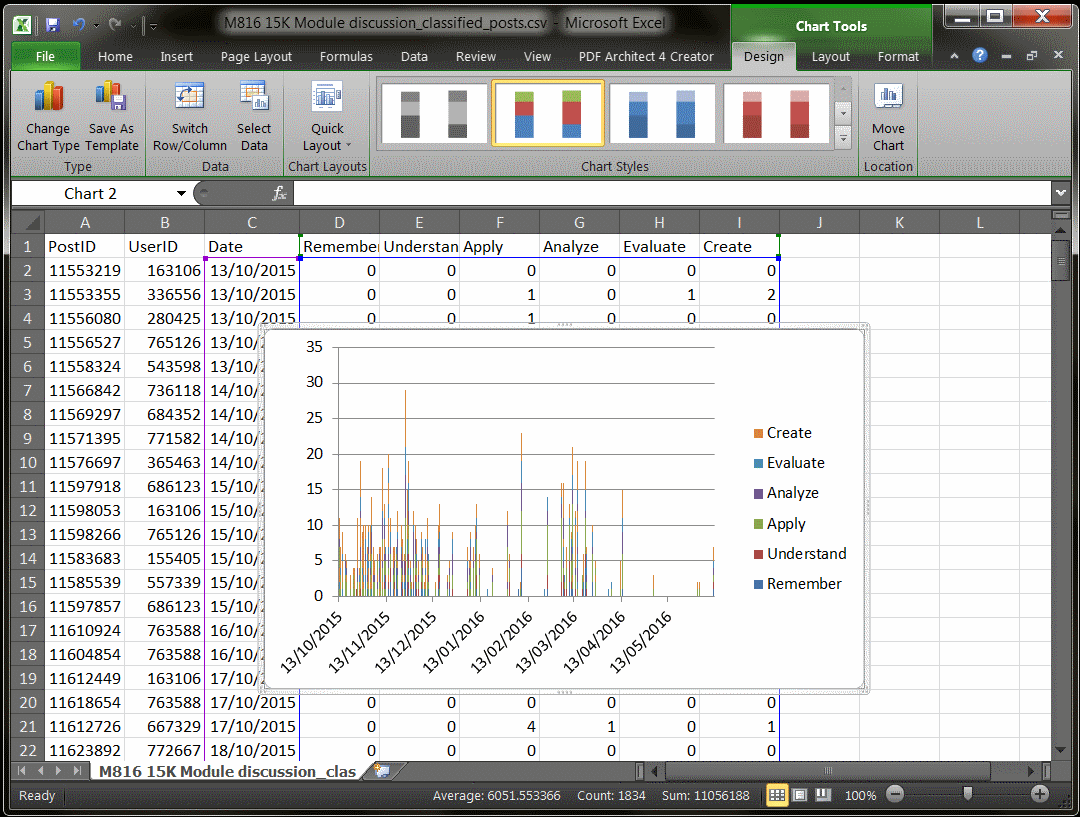


Figure 27: Stacked bar chart in Excel

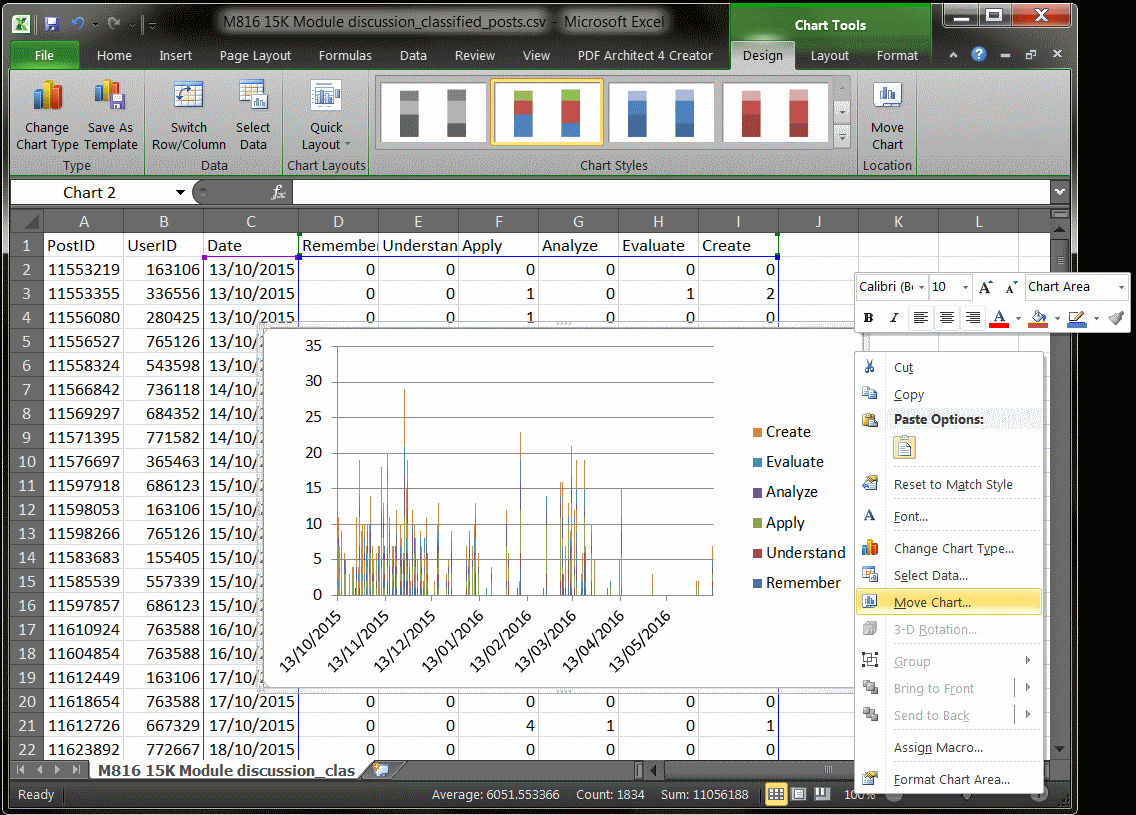


Figure 28: Move chart in Excel

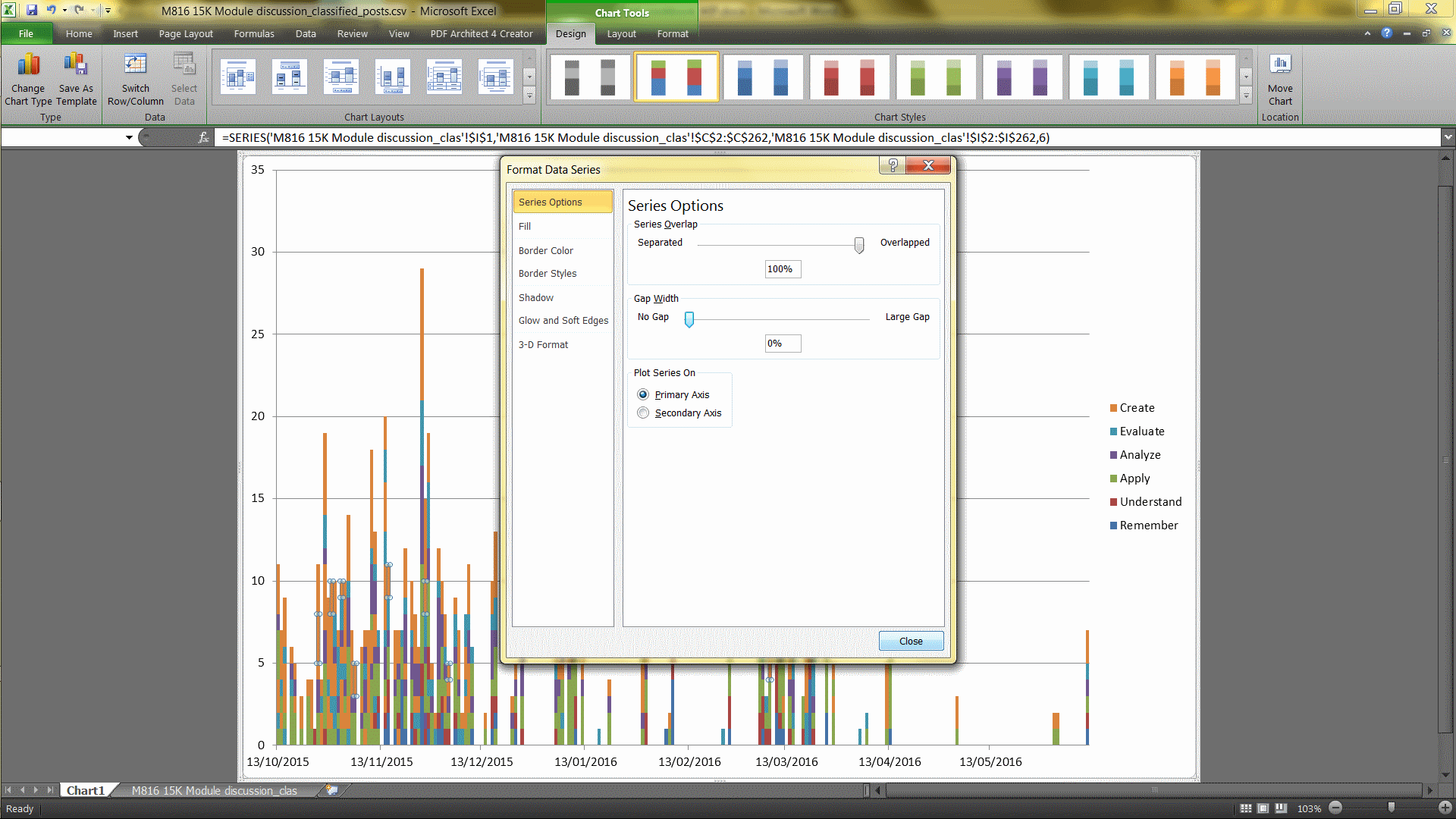


Figure 29: Format chart in Excel

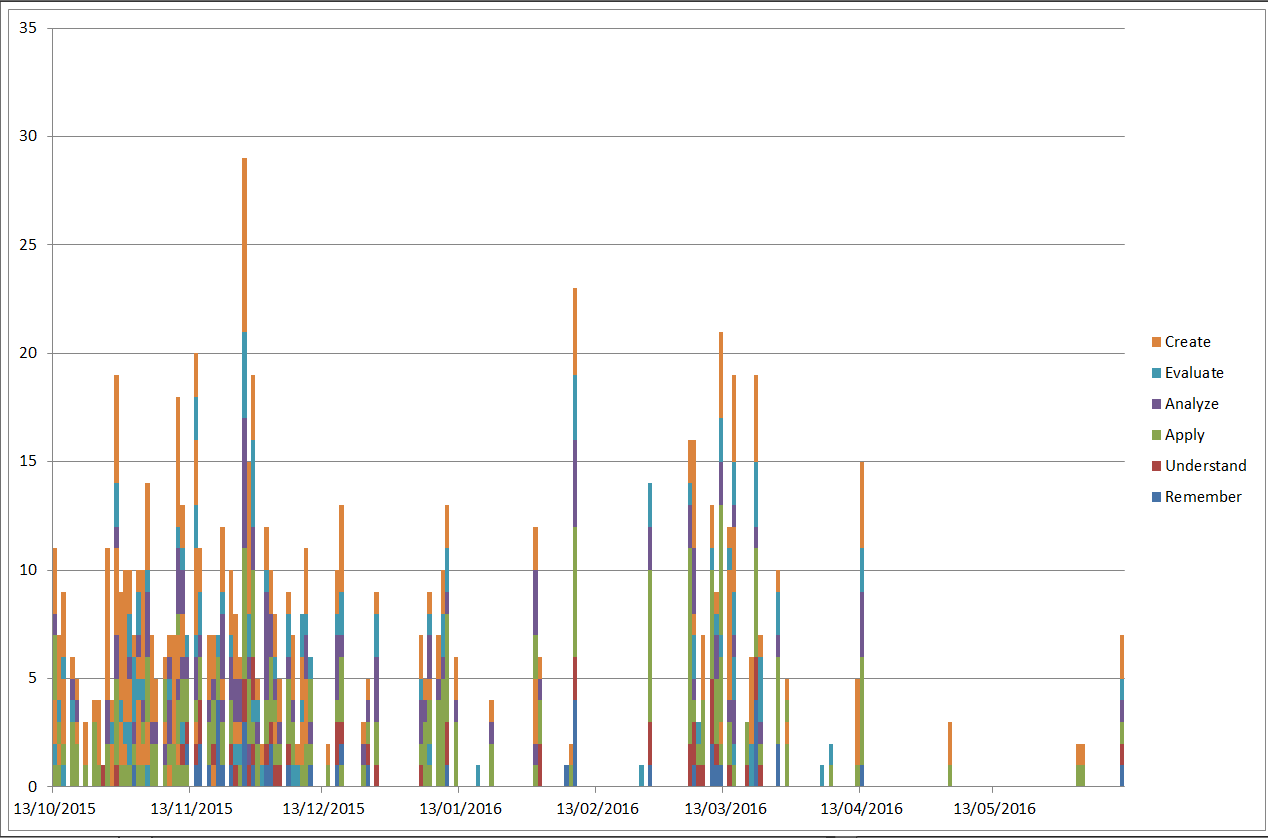


Figure 30: Formatted bar chart in Excel

The final chart lets you compare the nature of the posts when classified according to Bloom’s Cogntive Taxonomy. The blocks within each column are ordered to match the increasing cognitive demand defined by Bloom. This view of the data helps assess if the posts’ content matches that expected from the course activities by the course team. It enables to assess the impact of our teaching interventions by assessing if the students are responding as expected.

## Draw a stacked percentage chart of classified posts

This recipe draws a stacked percentage chart for each day of the course on which one or more posts are made. The chart shows the percentage use of each classification level in the posts. . This helps you assess the content of the posts, and see if the nature of that content changes.

It is very similar to the previous recipe, , but presents a different view of the data. By offering a different perspective, more insight may be gained regarding the impact of the teaching on students gauged by their forum posts.

This recipe uses Microsoft Excel. Open the file data/ M816 15K Module discussion\_classified\_posts.csv in Excel. See Figure 25: Classified posts opened in Excel on page .

Select the Date and Classification columns in the spreadsheet, columns C to I inclusive. From the Insert ribbon select the Column drop down menu and from that select the 2-D Column, 100% Stacked Column option. See Figure 31: Select 100% Stacked Column in Excel on page 23.

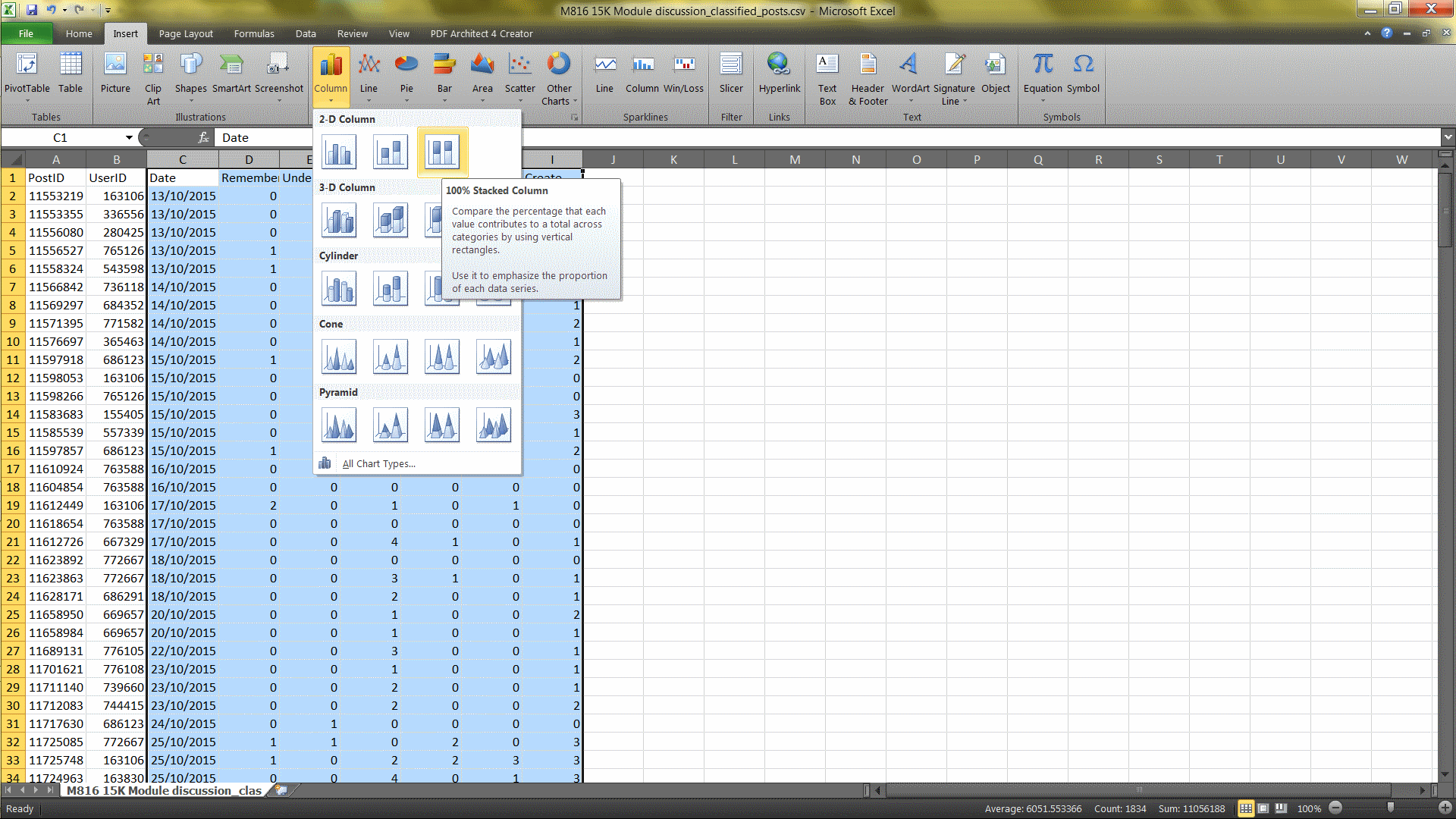


Figure 31: Select 100% Stacked Column in Excel

This produces the chart shown in Figure 32: Stacked percentage bar chart in Excel on page 24.

That chart can be moved and formatted as in the previous recipe to produce the final version as shown in Figure 33: Formatted percentage chart in Excel on page 24, and saved in data/ M816 15K Module discussion\_classified\_posts.xlsx.

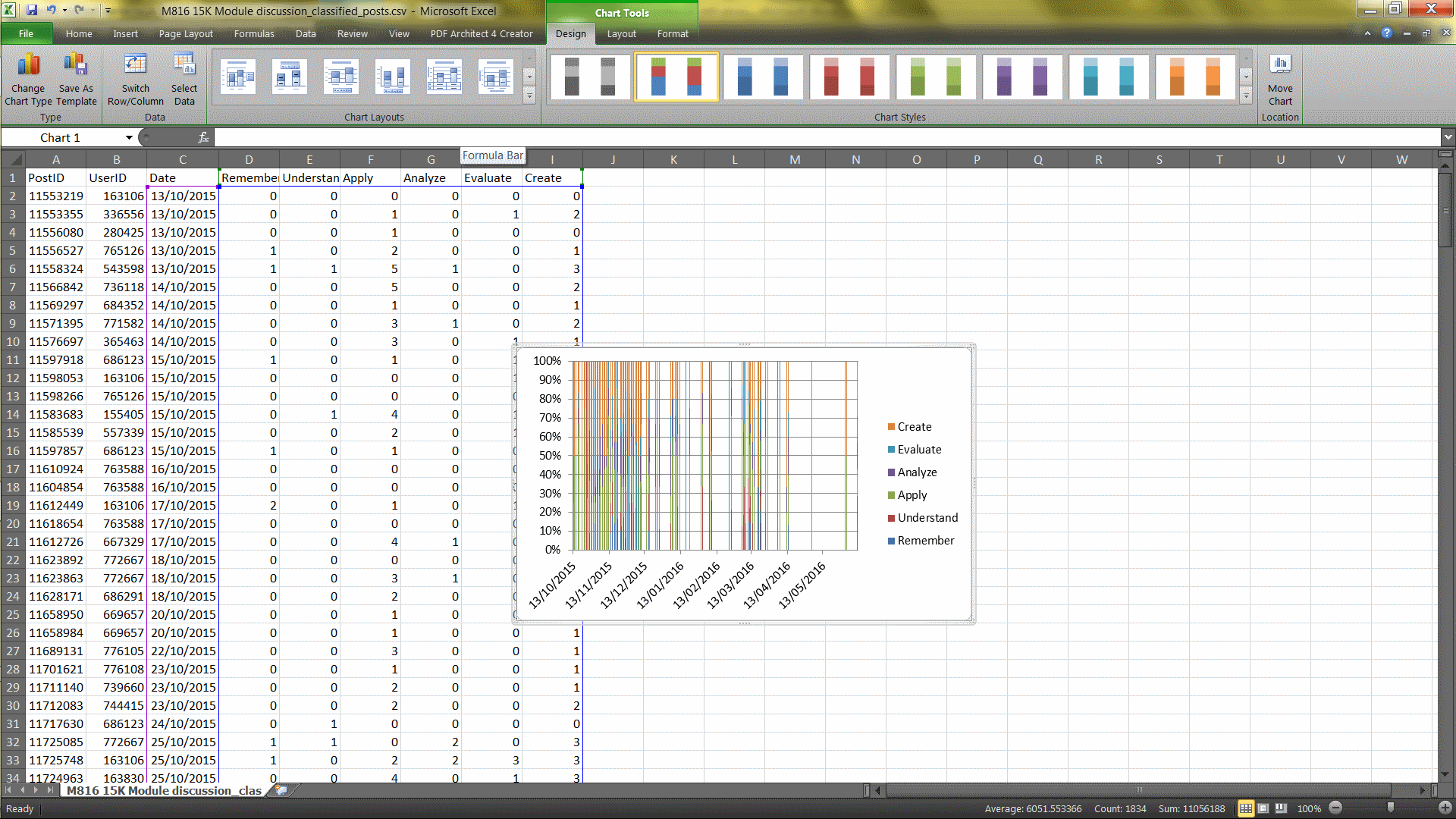


Figure 32: Stacked percentage bar chart in Excel

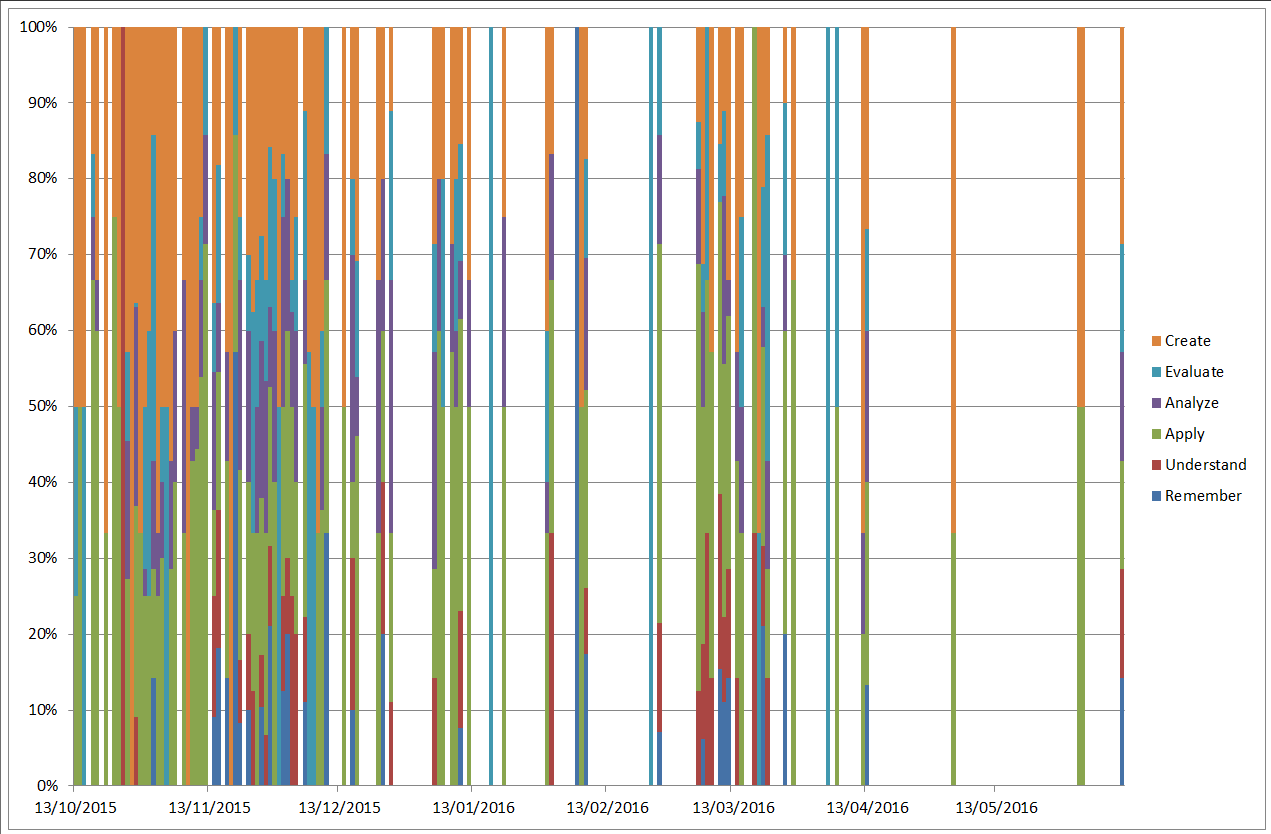


Figure 33: Formatted percentage chart in Excel

This presents the classifications as a percentage, and so you can assess their relative use in the posts. This makes it easier to assess changes in use during a course, and relate them to course activities.