An Introduction to Plotly for Patent Analytics

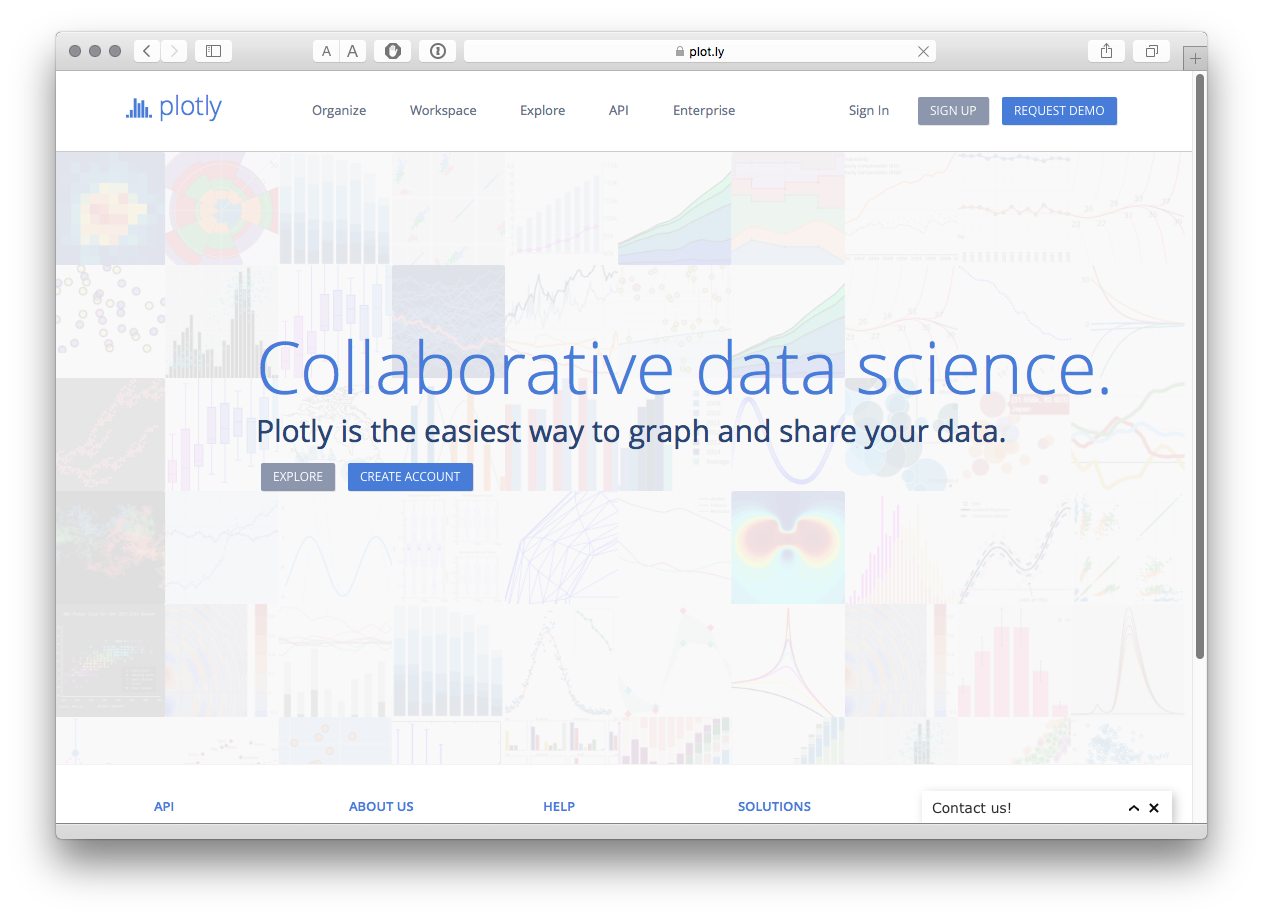
In this article we provide a quick introduction to the online graphing service [Plotly](http://plotly.com) to create graphics for use in patent analysis.

Plotly is an online graphing service that allows you to import excel, text and other files for visualisation. It also has API services for R, Python, MATLAB and a Plotly Javascript Library. This means that you can send data directly to Plotly for visualisation. Plots can also be shared with team mates or publicly.

Plotly's great strength is that it produces attractive interactive graphics that can easily be shared with colleagues or made public. It also has a wide variety of graph types including contour and heatmaps. For examples of graphs created with Plotly see the [public gallery](https://plot.ly/feed/). However, we experienced significant difficulty in loading patent data into Plotly and we struggled to draw graphs that would take a couple of minutes if we used Tableau Public as our benchmark. This suggests a need for investment of more time in understanding the expectations of the tool to make best use of the service. It may also suggest that while Plotly can produce excellent graphics, the service may need to invest more time in explaining how ordinary human beings might format their data to meet its expectations. In short, as is often the case with open source tools, the basic documentation could be better. Do not let this put you off. Plotly is potentially a great tool for creating and sharing graphics. In the rest of this article we will help you get started as a basis for exploring Plotly on your own.

### Getting Started with Plotly

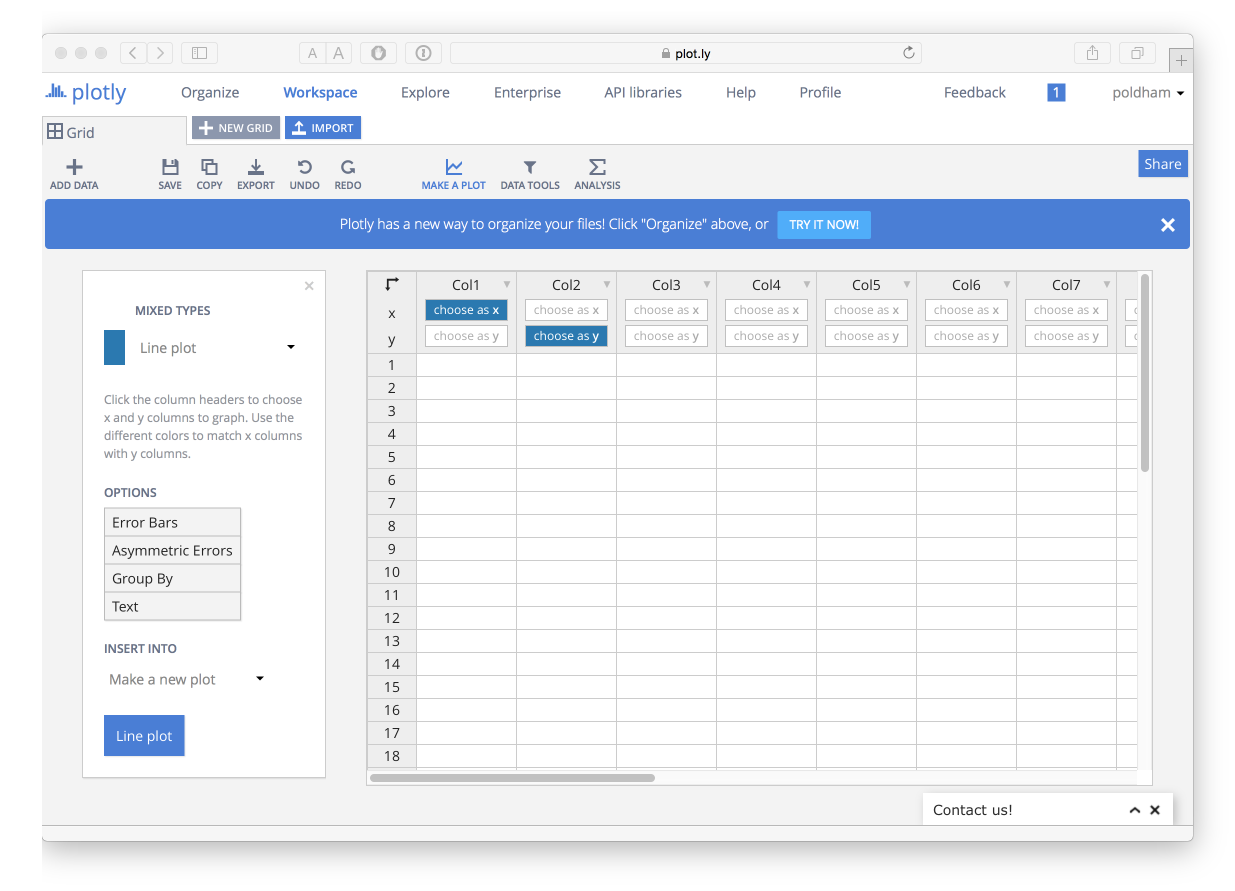
We need to start out by creating an account using the Create Account Button.



Next we will see an invitation to take a tour (which is worth doing) and Plotly helpfully points out that we can load files from Google Drive or Dropbox. We then select the Workspace option to begin work.

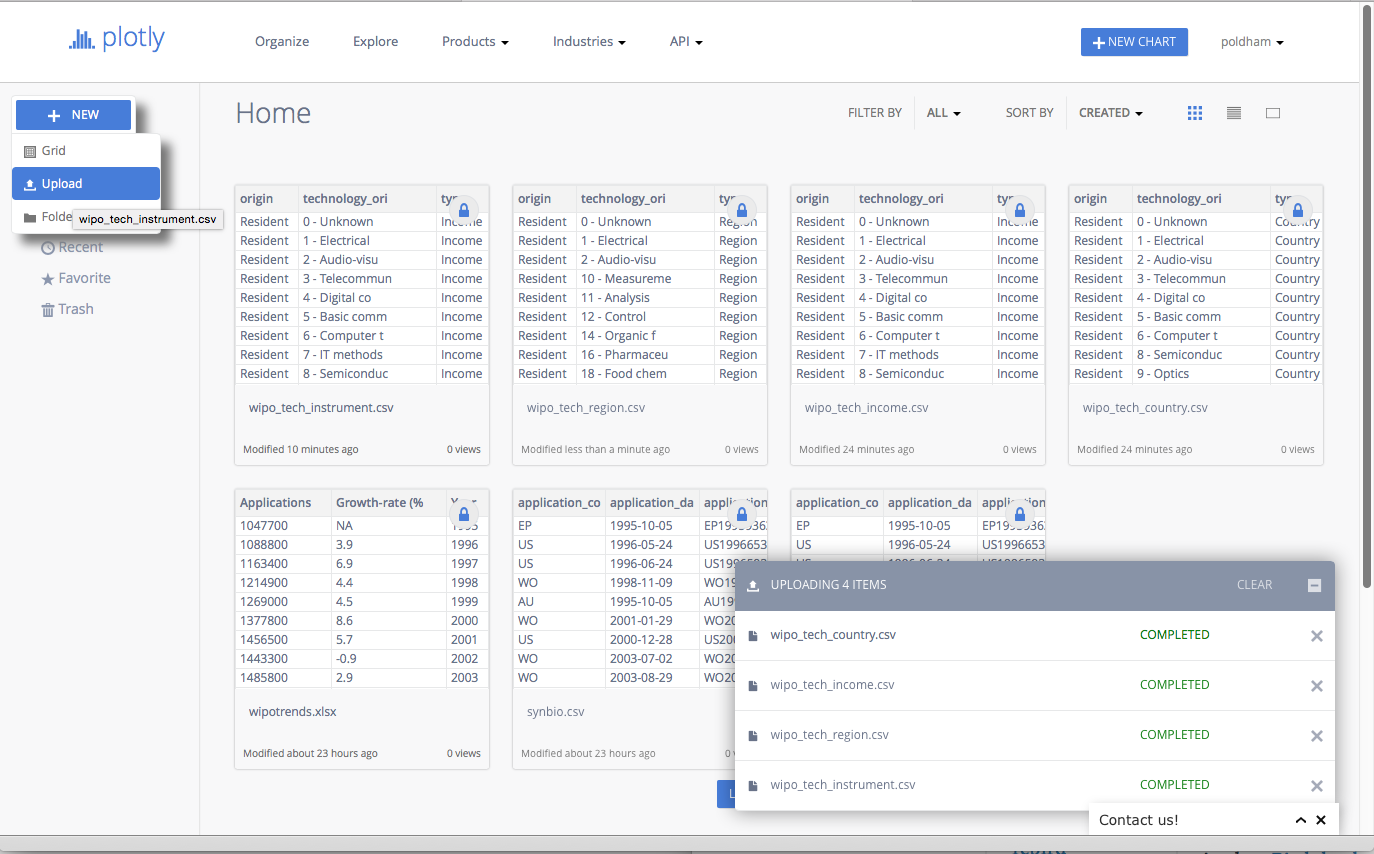
### Importing Files

When you first arrive you will see a Workspace with a Grid (Plotly's term for a table or worksheet).



In the workspace you will see an Import Icon that provides a range of options for importing data. Don't import anything yet! You can also copy data from a file and paste it into the Grid.

The reason not to use these options at the moment is that while the data may import fine first time, in other cases it will not. Using the options on this page you will receive no information if an import fails. We also encountered problems with saving data that had been pasted into the worksheet (even where it appeared to work). To avoid potential frustration head over to Organize.



From the Organize page select the New button and then Upload. Now select your local file. When you upload the file a status message will display and if all goes well then you will see a completed message. If not a red message will display informing you that there has been a problem (how you fix these problems is unclear)

For this experiment we used tow datasets from the Open Source Patent Analytics Manual [data repository](https://github.com/poldham/opensource-patent-analytics/tree/master/2_datasets). When using the Github repository click on the file of interest until you see a View Raw message. Then right click to download the data file from there. You can download them for your own use directly from the following links.

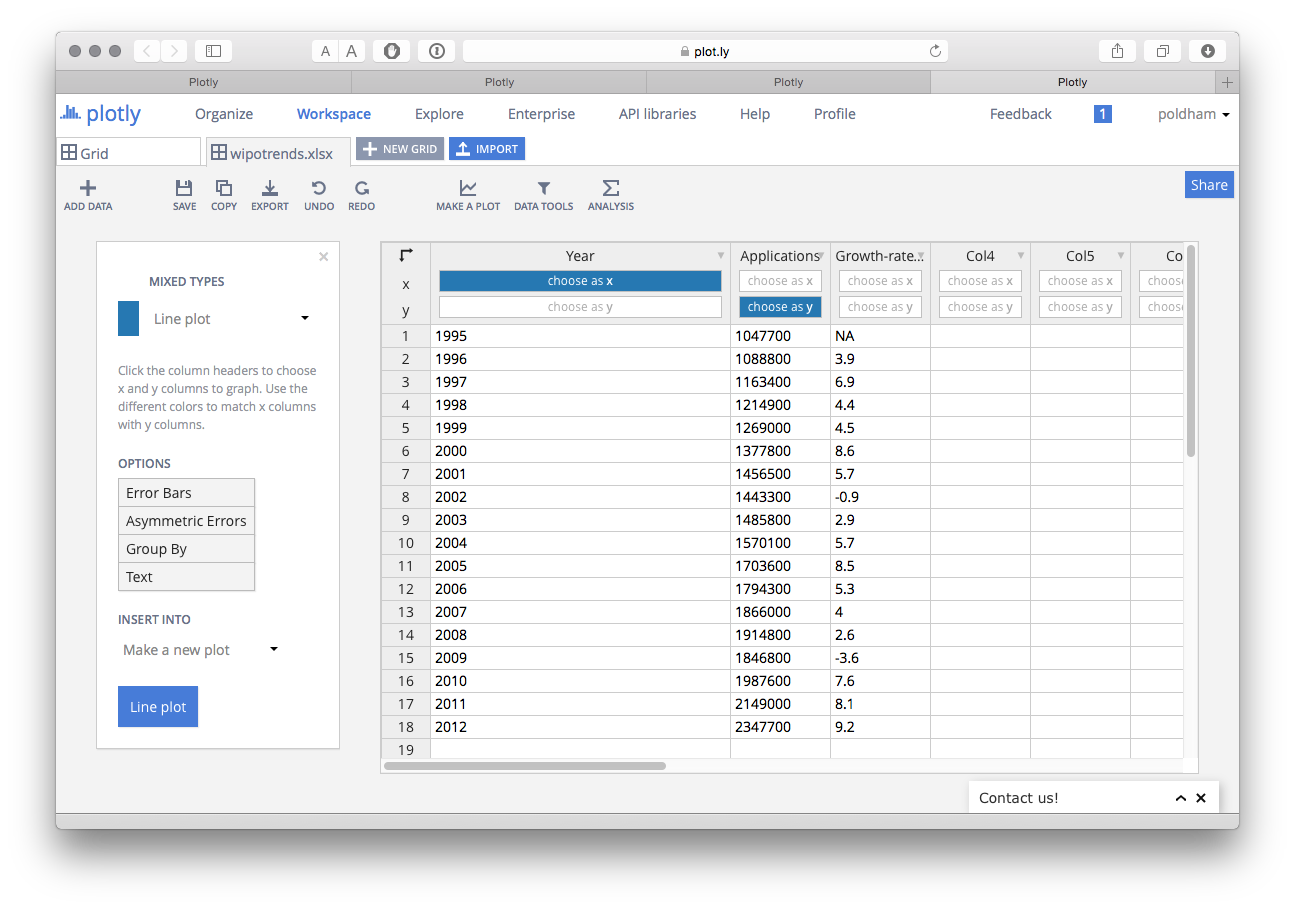
1. [WIPO trends](https://github.com/poldham/opensource-patent-analytics/blob/master/2_datasets/wipo/wipotrends_cleaned.csv?raw=true) application trends by year and with % change
2. [Pizza patents by country and year](https://github.com/poldham/opensource-patent-analytics/raw/master/2_datasets/pizza_medium_clean/pcy.csv). This is a simple dataset containing counts of patent documents containing the word pizza from [WIPO Patentscope](https://patentscope.wipo.int/search/en/search.jsf) broken down by country and year.

One important point to note is that Plotly is not a data processing tool. While there are some data tools, your data will generally need to be in a form that is suitable for plotting at the time of input. In part this reflects the use of APIs which allow for users of Python, R and Matlab to send their data to Plotly directly for sharing with others. This is one of the great strengths of Plotly and we will cover this below. However, we also experienced problems in loading and graphing datasets that were easy to work with in Tableau (as a benchmark). This suggests a need to invest time in understanding the formats that Plotly understands.

We experienced a different type of problem with the simple WIPO trends data where Plotly concatenated the first row (containing labels) and the first data row into one heading row. However, in most cases import seemed to be fine. To turn a row into a heading row try right clicking the row with the headings and right clicking on use row as col headers. Then right click again to remove the original row.

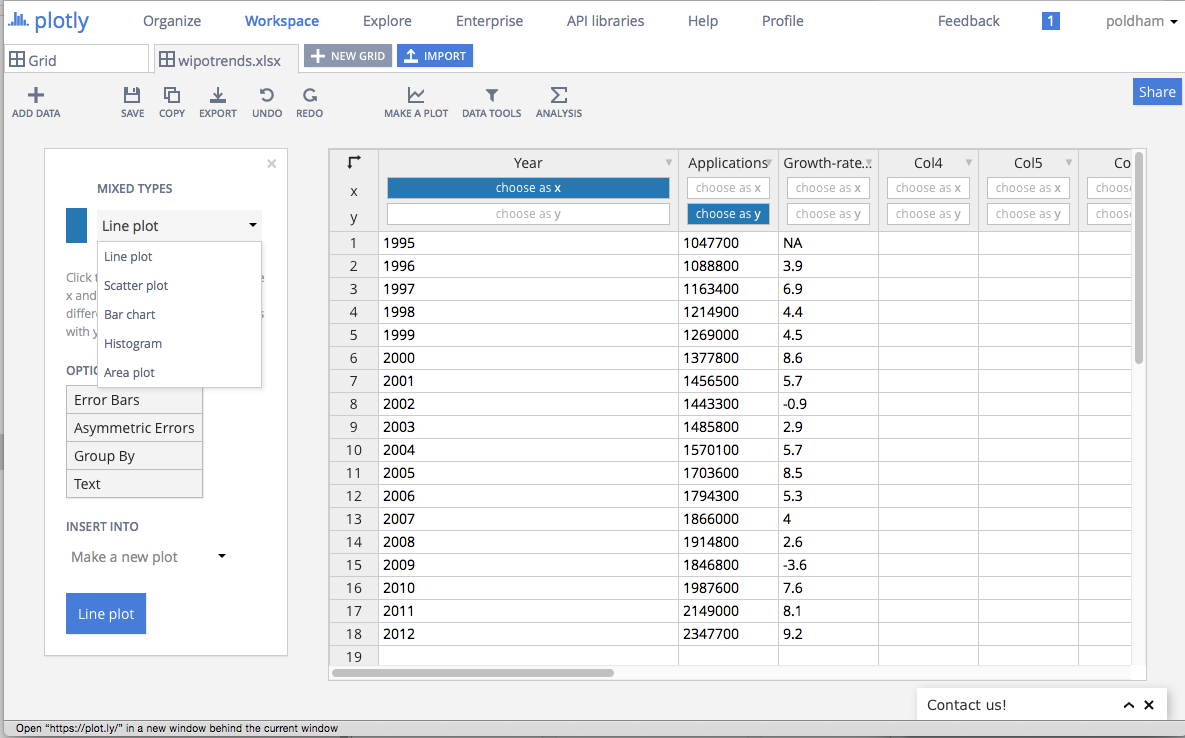
## Creating a graphic

We will start with the simple WIPO trends data by opening up that Grid

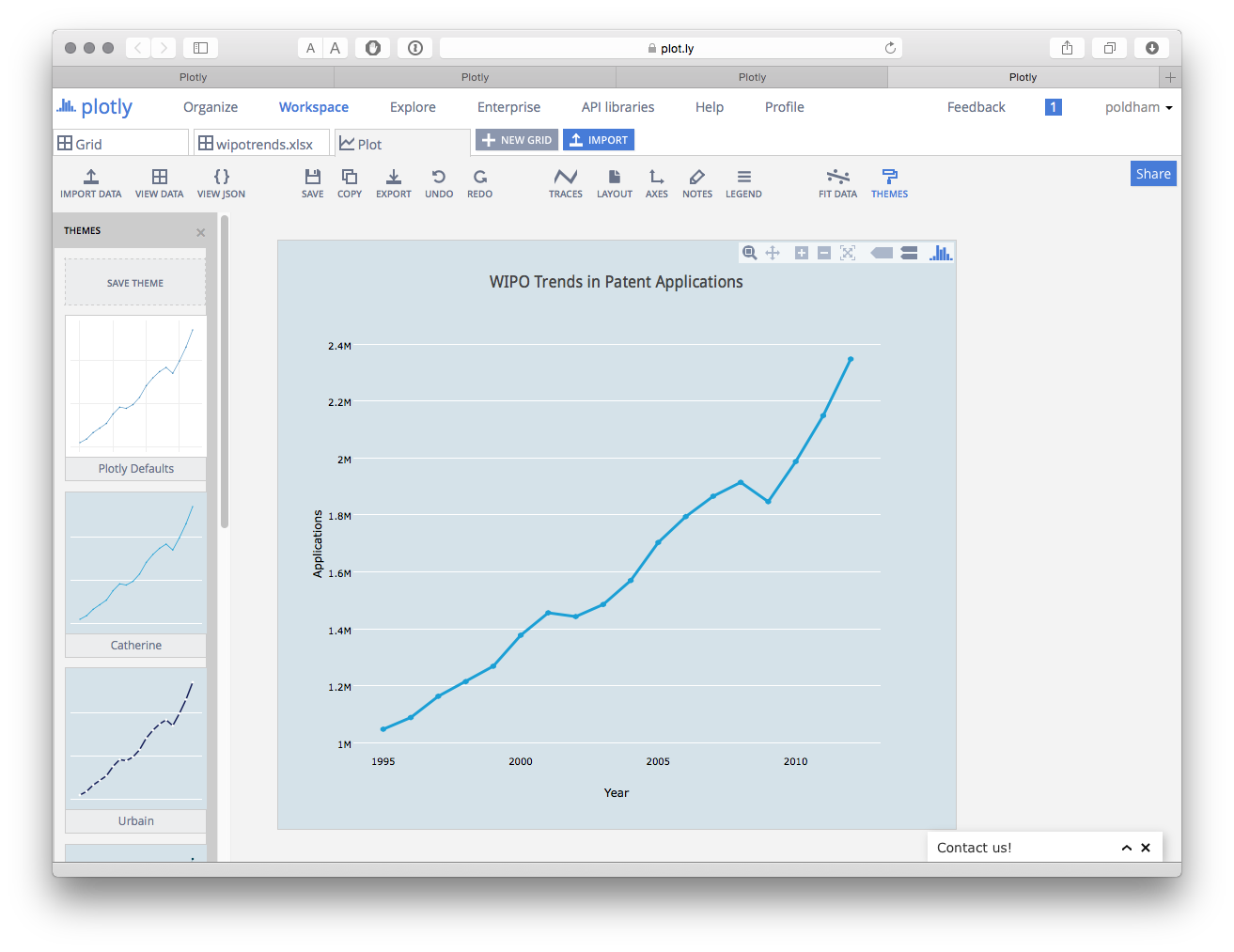


Note here that in the Grid we have options to select the x or y axis for plotting. There is also an Options menu that we will come back to.

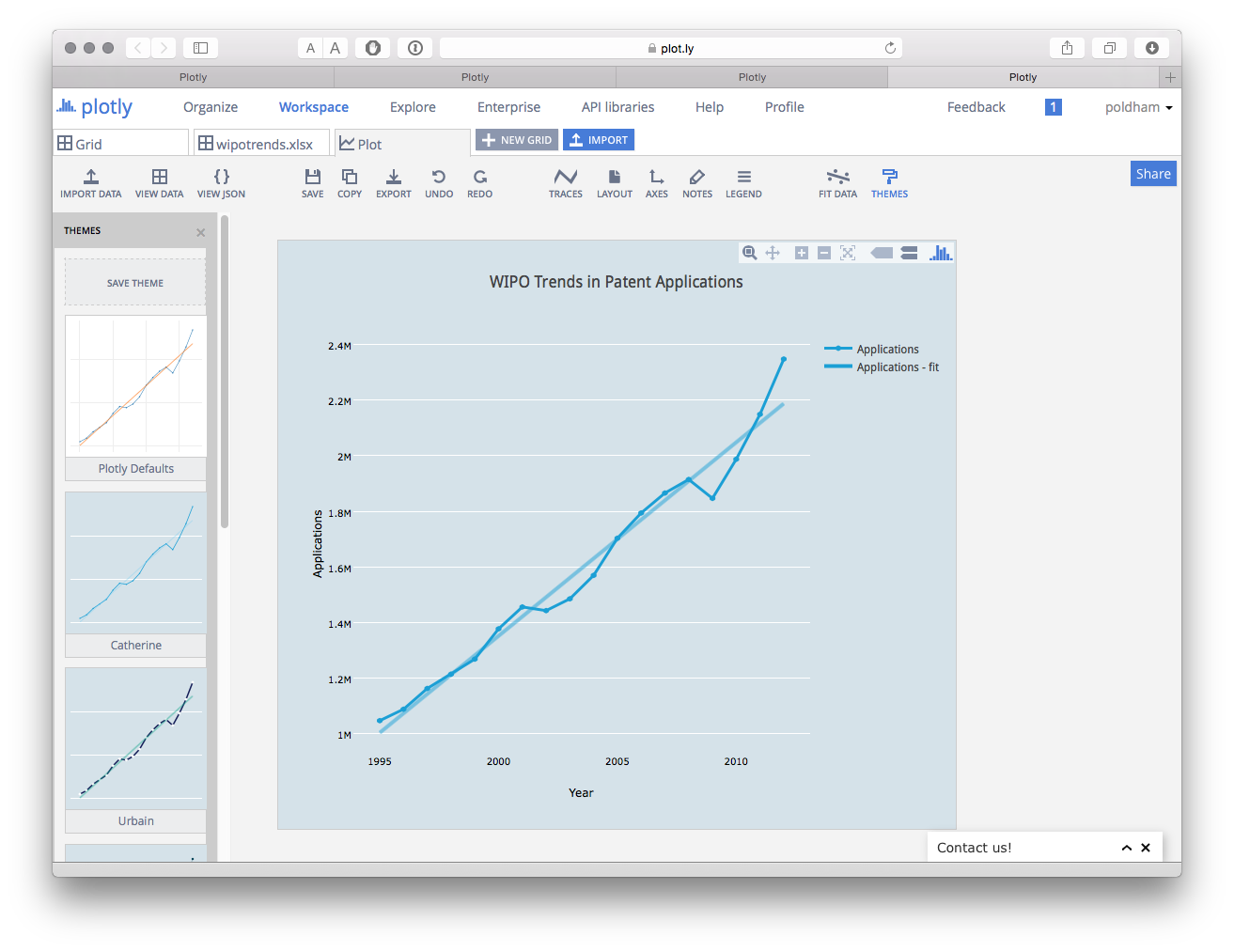
The Type of plot can be changed by selecting the drop down menu as we can see below.



Sticking with a line graph, when we create the plot we can add a title and then change the theme (in this case to Catherine).



We could also add a fit line by selecting the FIT DATA menu icon. This will ask you to create a fit and then you have a range of preset functions or you can add your own. Here we have simply chosen the default Linear fit.

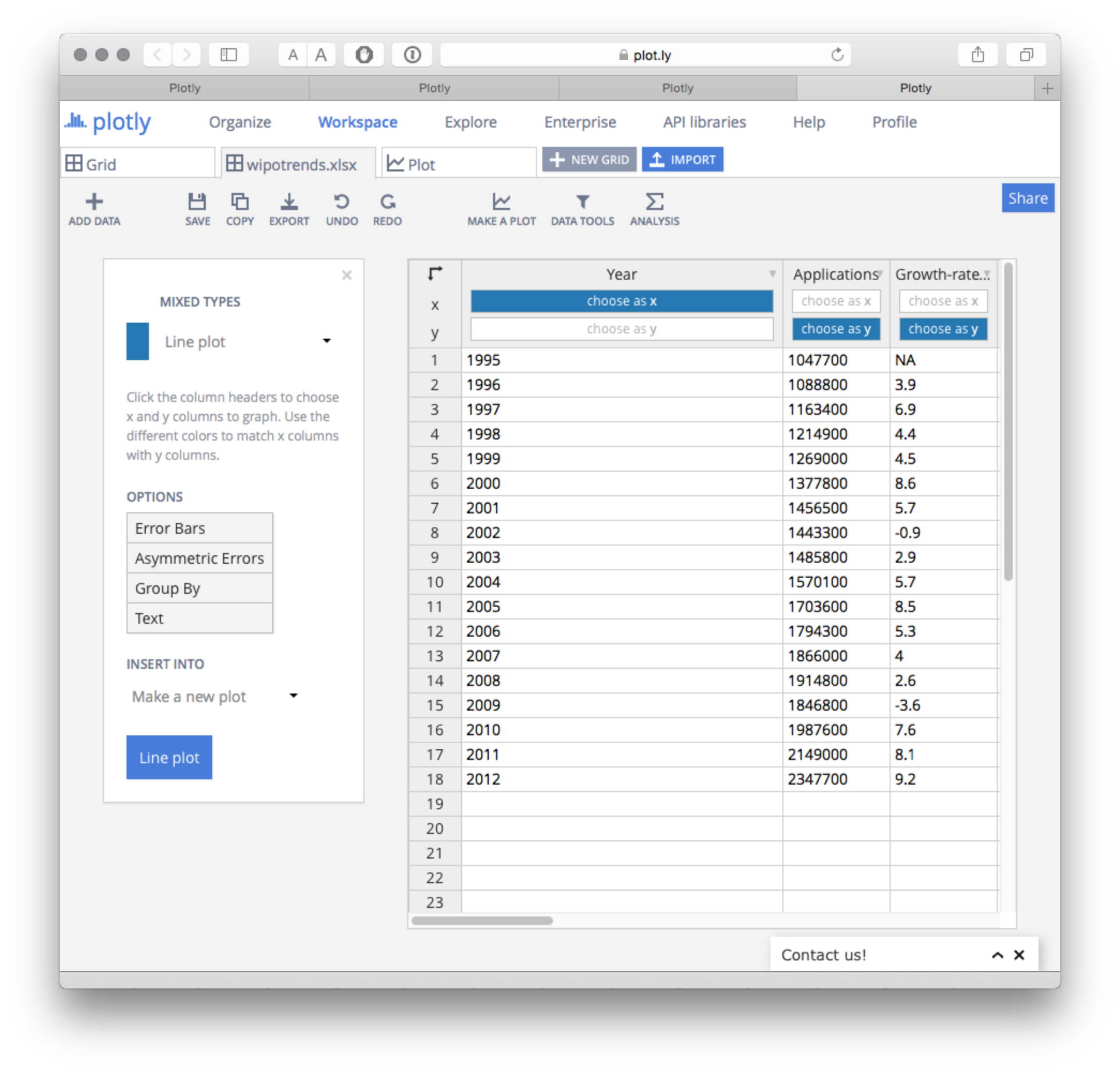


We can then save the plot and use the export button to save the plot in a variety of formats and sizes. It is also very easy to add annotations using the Notes icon. Confusingly, the large blue Share button only seems to save the file and despite saving the plot we were not able to locate it again. While Plotly certainly looks nice, and appears to have attractive functions it is not intuitive and the difficulties involved in importing and sharing can be frustrating and time consuming. In short, time is needed to invest in and explore the potential of this tool.

### Adding a Second Axis

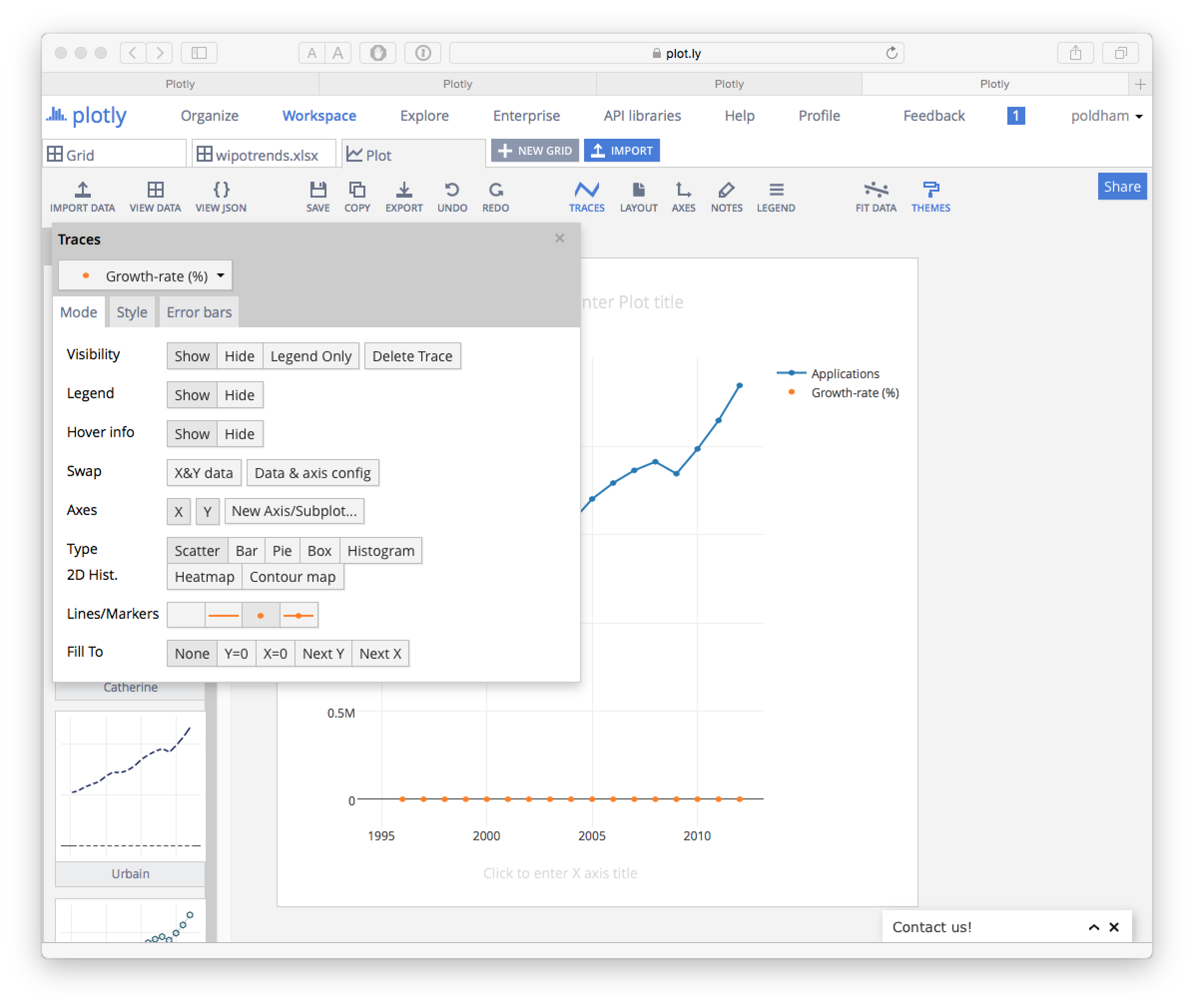
If we go back to our original WIPO trends data we have a percentage score for the year on year change in patent applications. We might want to show this on a plot with a second axis for the percentage.

To do that select the percentage as a second item for the y axis.

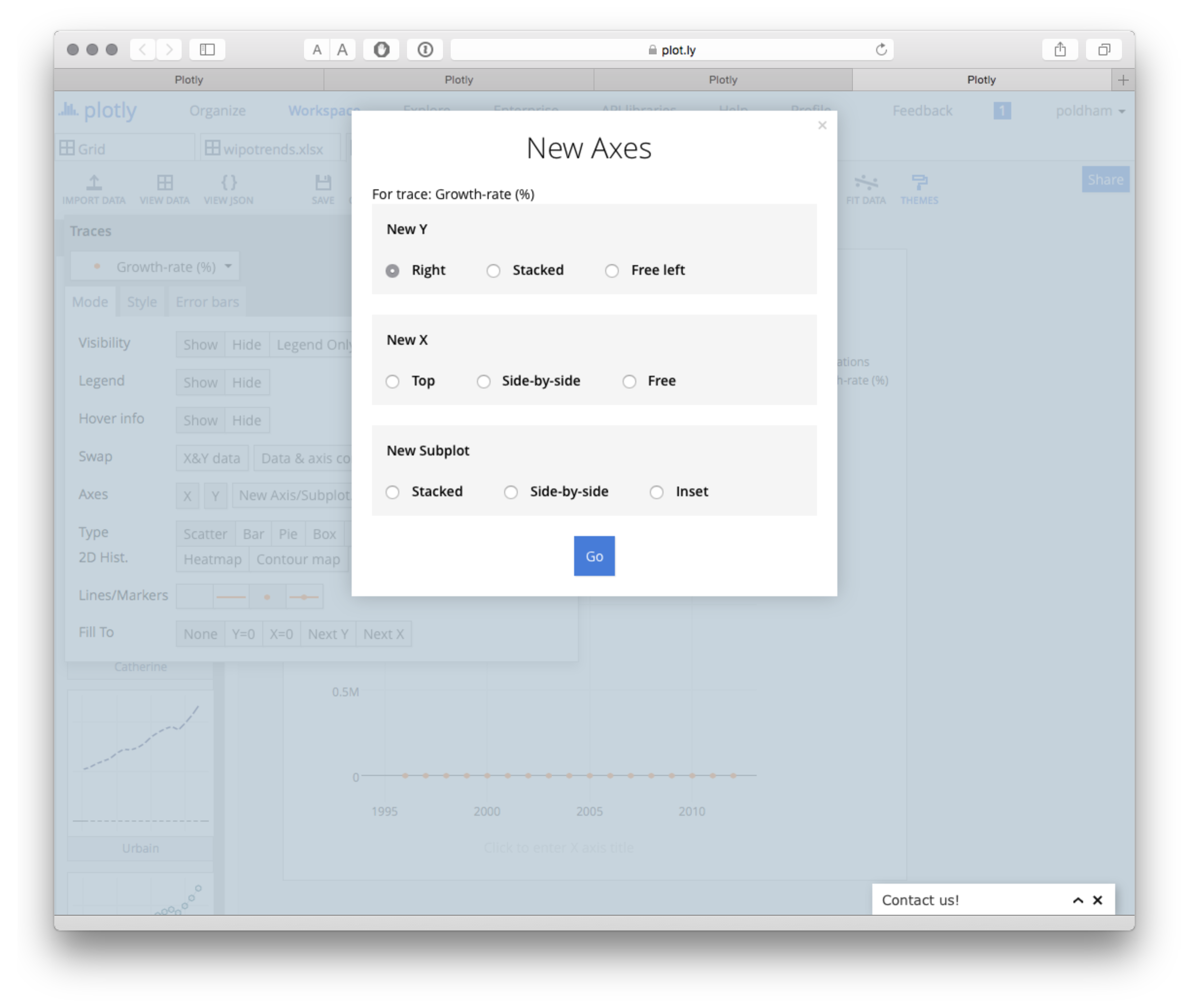


When we choose Line plot we will now see the two sets of data with the percentage trailing on the bottom. We now need to create a second y axis on the right and assignee the percentage data to that.

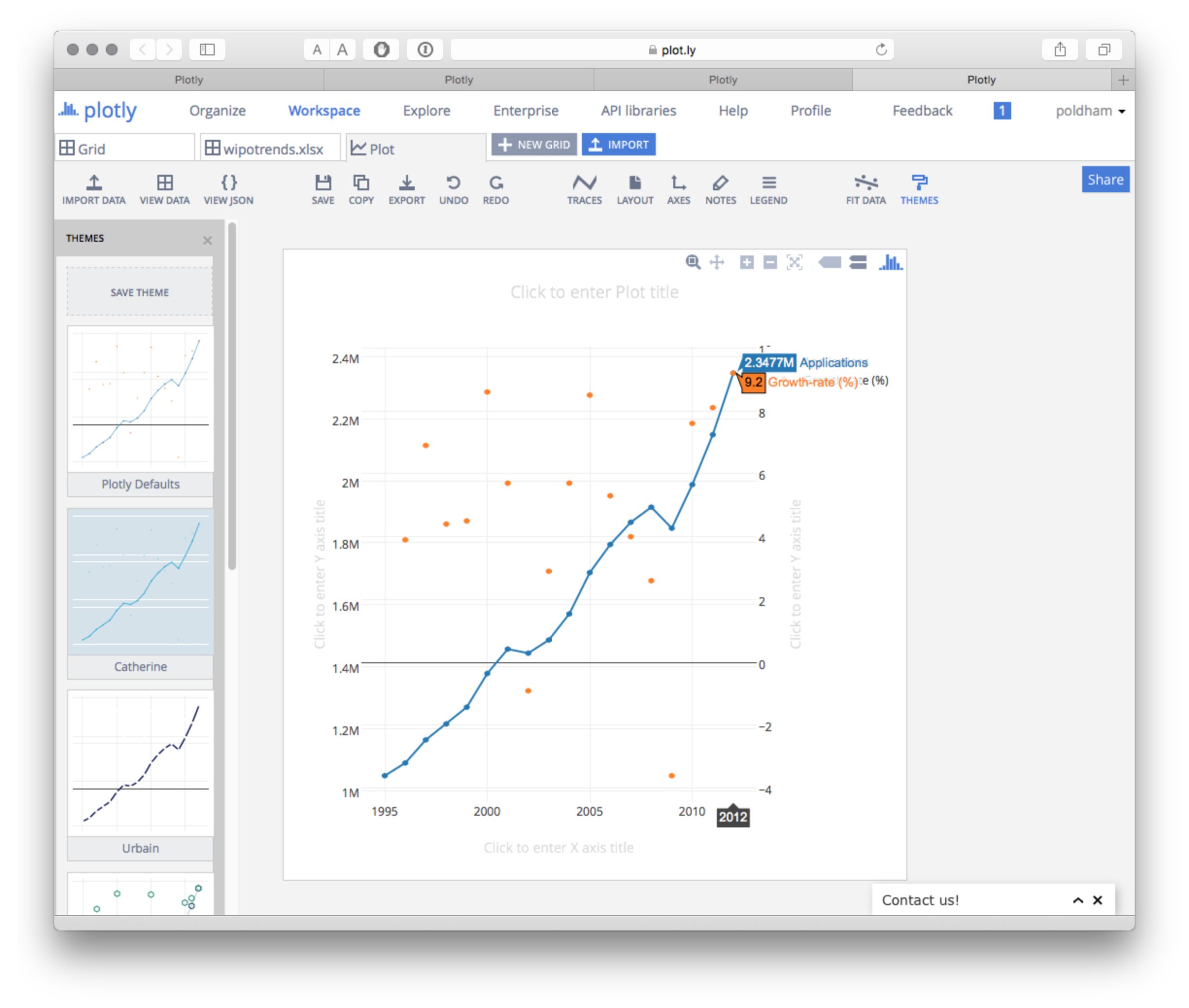
To do this select the Traces icon and a menu will pop up showing the data traces. Select Growth Rate % from the Traces menu. Then where you see Lines/Markers select the dot. This will prevent the percentage scores displaying as a line.



Next, select New Axis/Subplot and a new screen will pop up. We have some choices here but will simply choose to create a new axis on the right.

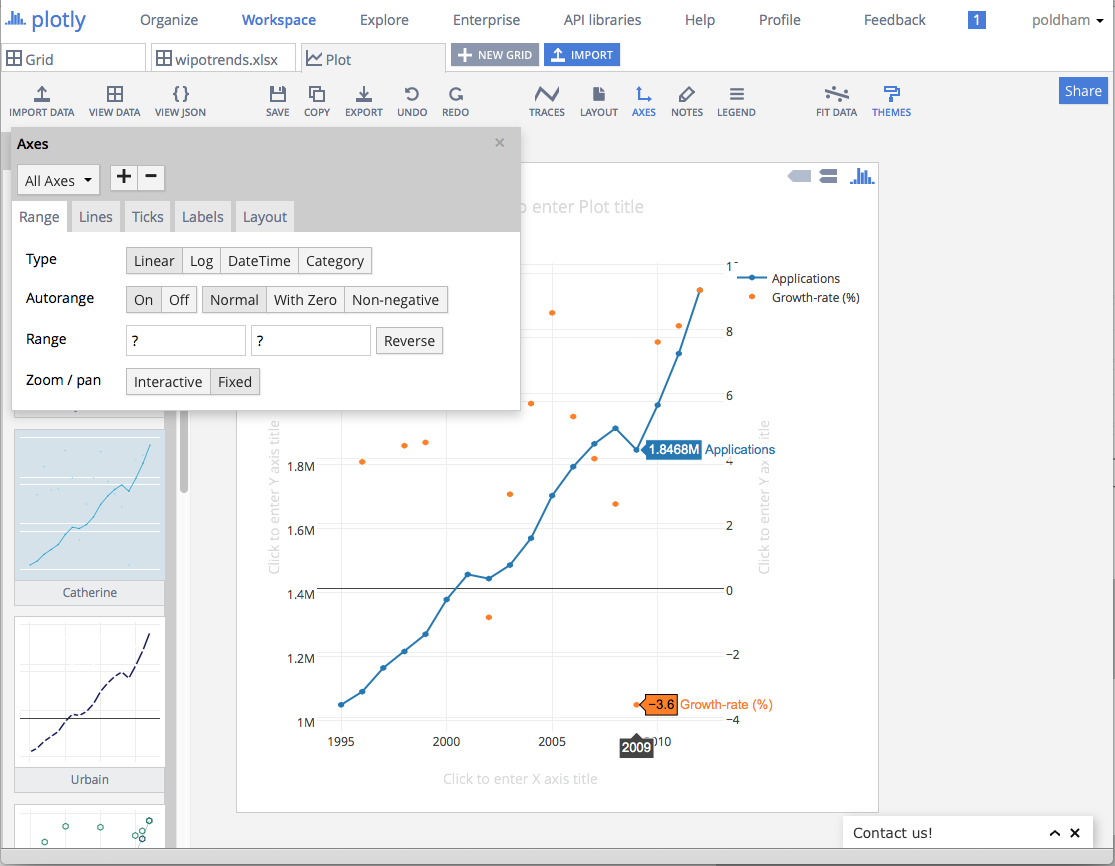


The result will look something like this.



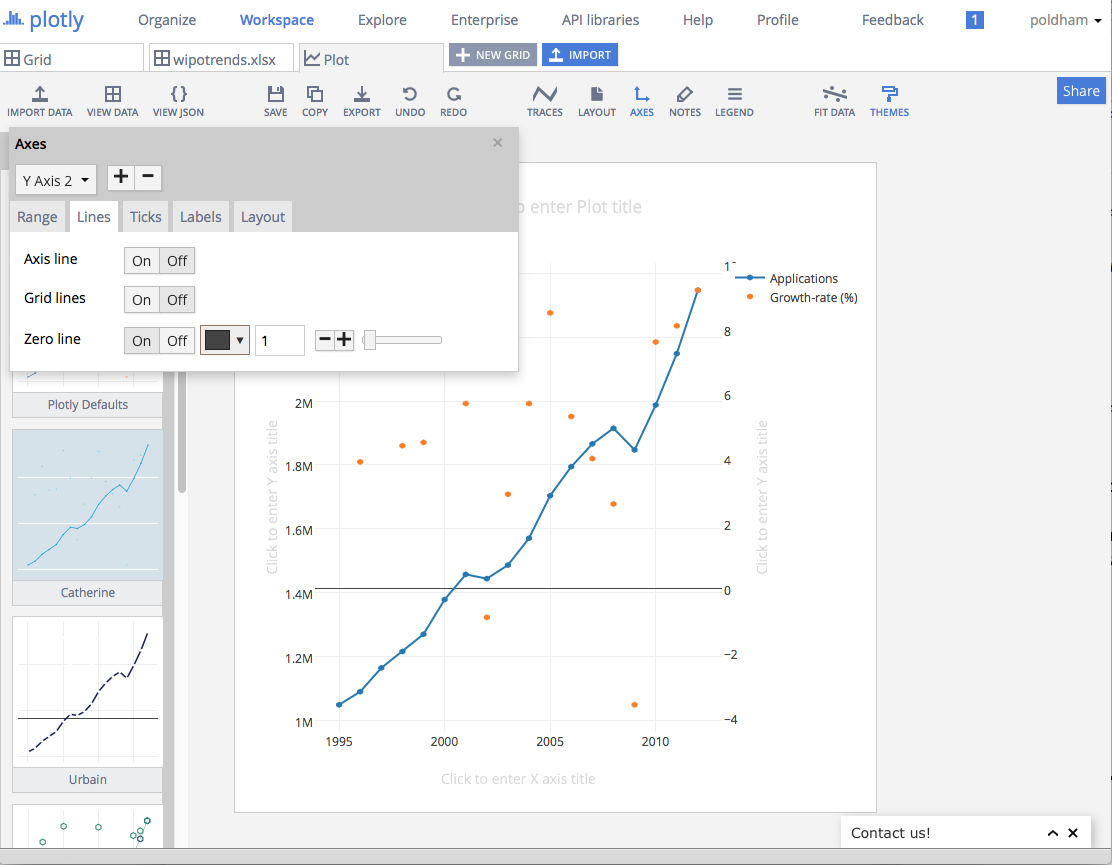
Our issue now is equalising the axes and changing the size of the points for the percentage scores. Finally we can add a title.

Before we go any further let's note that we have a significant minus axis value of -3.6% in 2009 when patent applications declined. There is also a minus value in 2002.

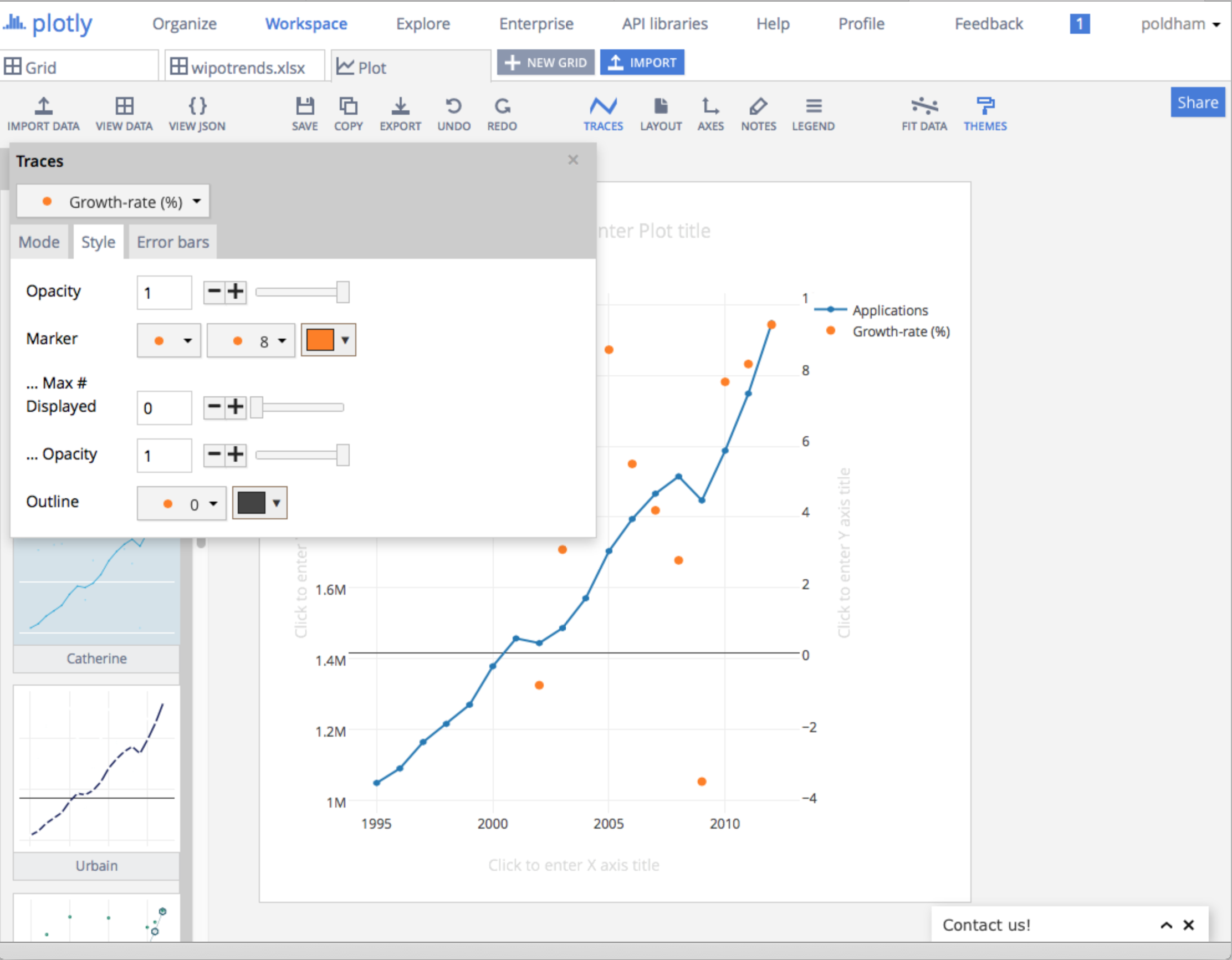


If we wanted to retain these values we would probably want to turn off the second set of grid lines. We would also want to resize the points.

To turn of the grid lines on the second y axis Click on the Axes icon and then from the All Axes drop down select Y Axis 2, then Lines and Grid lines OFF. Also turn the Zero line to off unless you want to retain it.



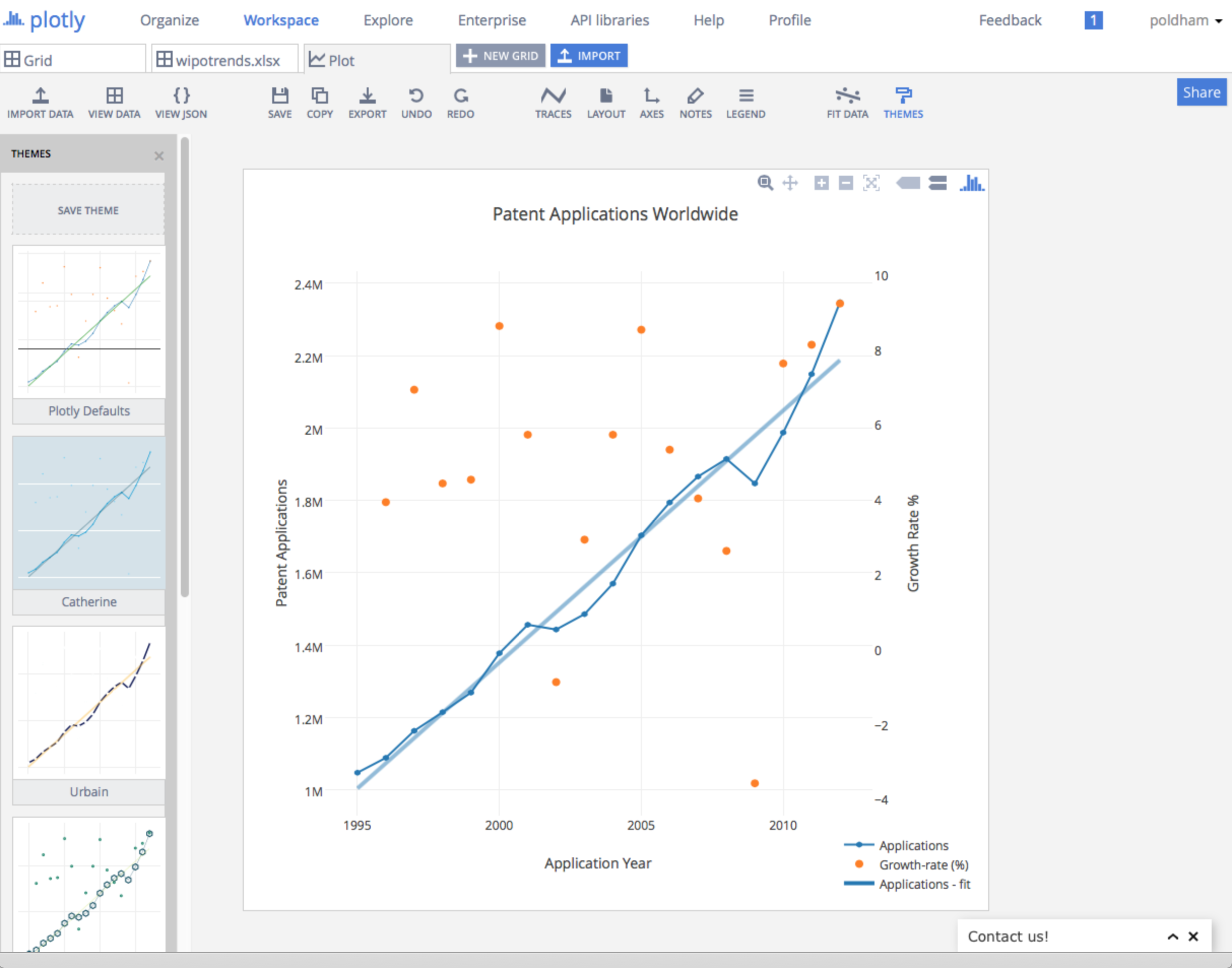
To resize the points we need to go back to traces and select Growth Rate from the list of Traces. Then choose the Style tab and change the marker size to something larger such as 8.



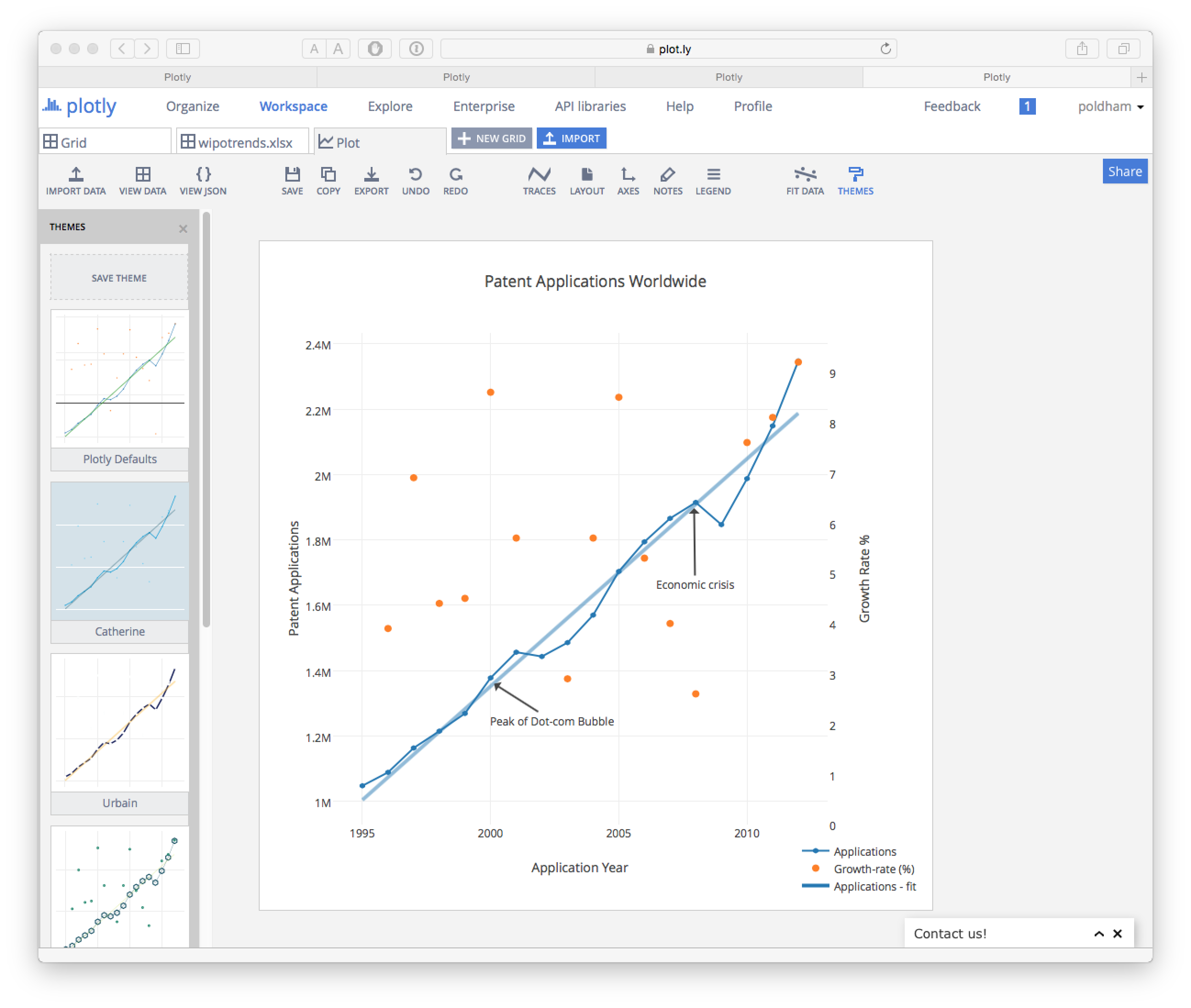
We can simply type in the Axis labels and a title into the text boxes provided. By choosing the Legend icon we could turn the legend on or off. Note that while this graph could be seen as self explanatory it may not be for the reader. We can also simply drag the axes labels to a different position.

It is possible that we would want to remove the negative values from the plot (in that case the values would need to be explained in the accompanying text). To do that select Axes, then Y Axis2 then in Autorange choose Non-negativeto show only values over zero on the plot.

If we wished we could also apply a fit line by choosing the FIT DATA icon. We will choose Linear.



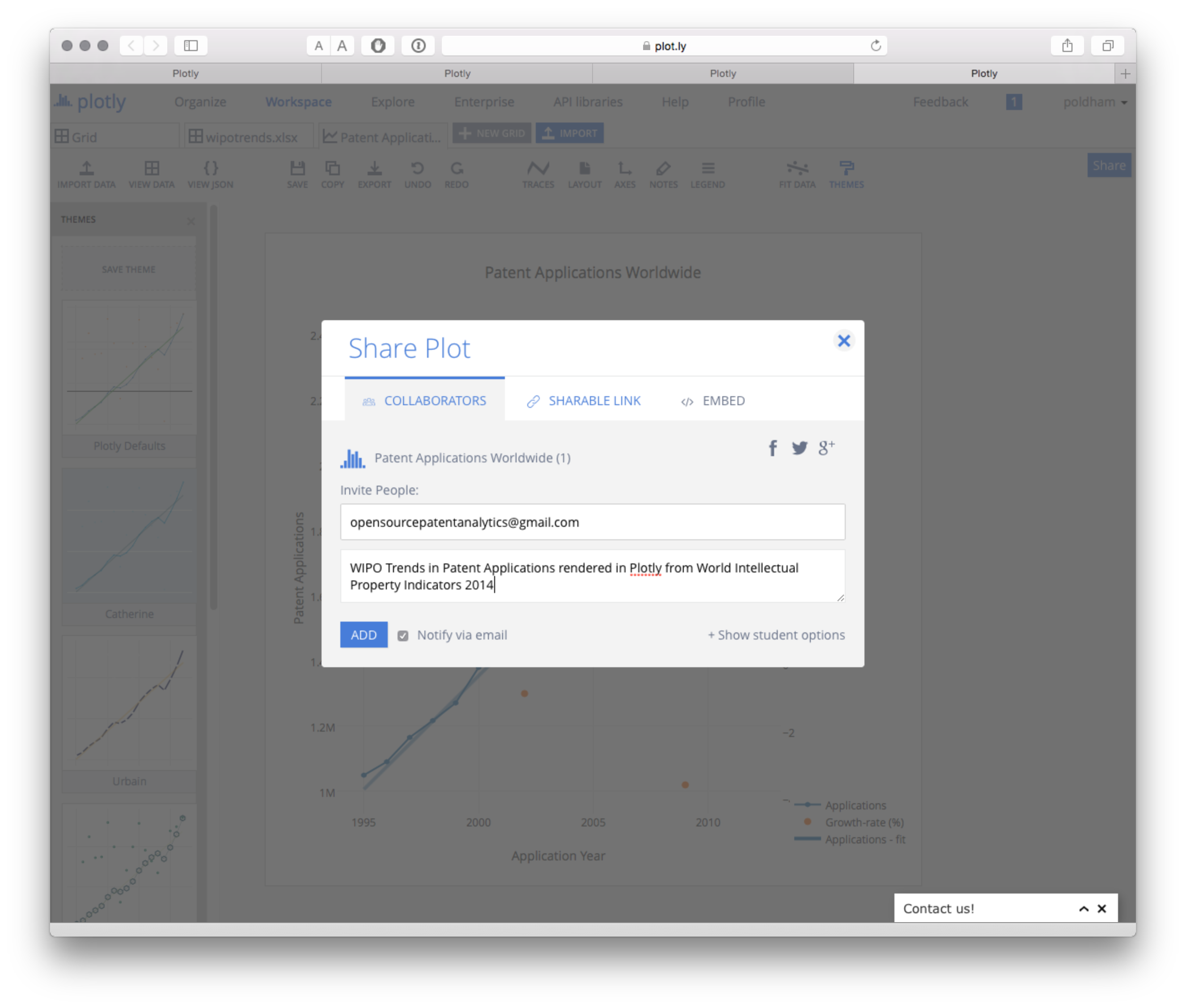
Finally, to finish off the plot we might want to add annotations using the NOTES icon. Simply click on the plus sign in the pop up menu for a new annotation and then select the arrow and text and move it into the position you want.



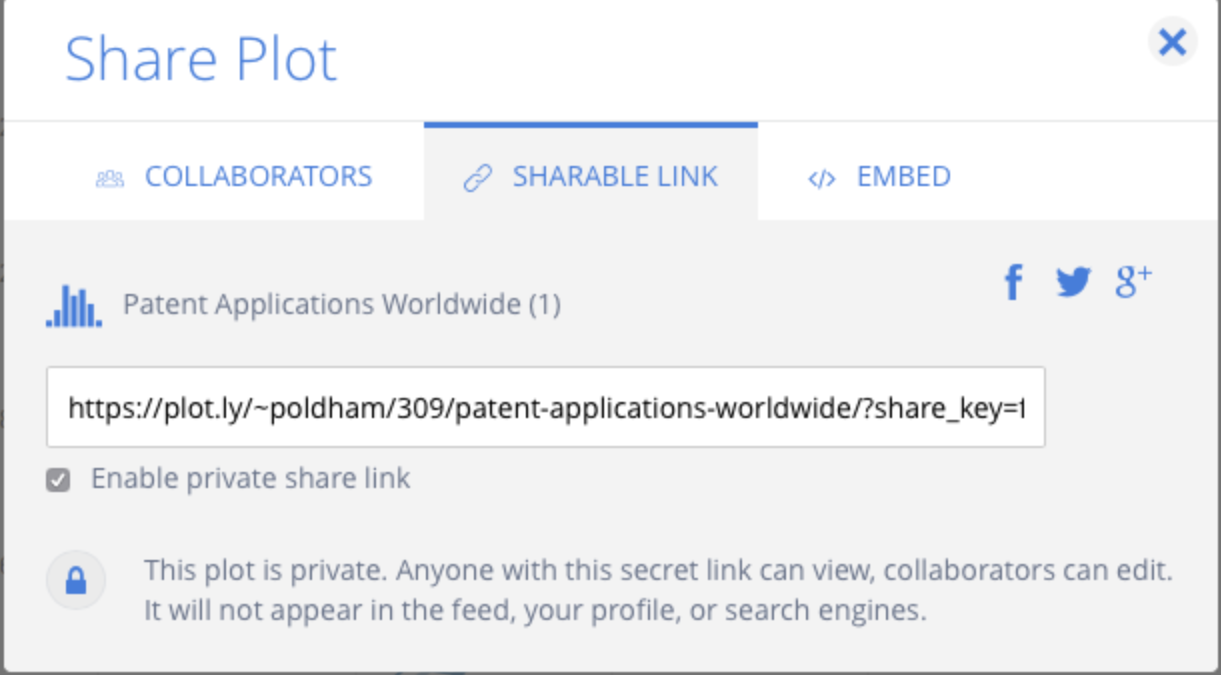
In this case we have added a couple of markers that may help to understand trends in activity. First, we have a dip in patent applications between 2001 and 2002. One possible explanation here is that this is a knock on effect of the collapse of the dot.com bubble where share prices reached a peak in 2000, declined rapidly and recovered before declining again into 2001. Patent data typically displays lag effects and it is reasonable to think that the decline in application activity from 2001 reflects these wider economic adjustments. Similarly, there is a significant dip in applications between 2008 and 2009 that it appears reasonable to assume reflects the knock on effects of the global economic crisis of 2007-2008. Note here that these are grosso modo way markers. We could choose to add other timeline style events or layer graphics to help understand the potential or actual relationships between wider economic activity and trends in patent applications worldwide.

### Saving and Sharing Sharing

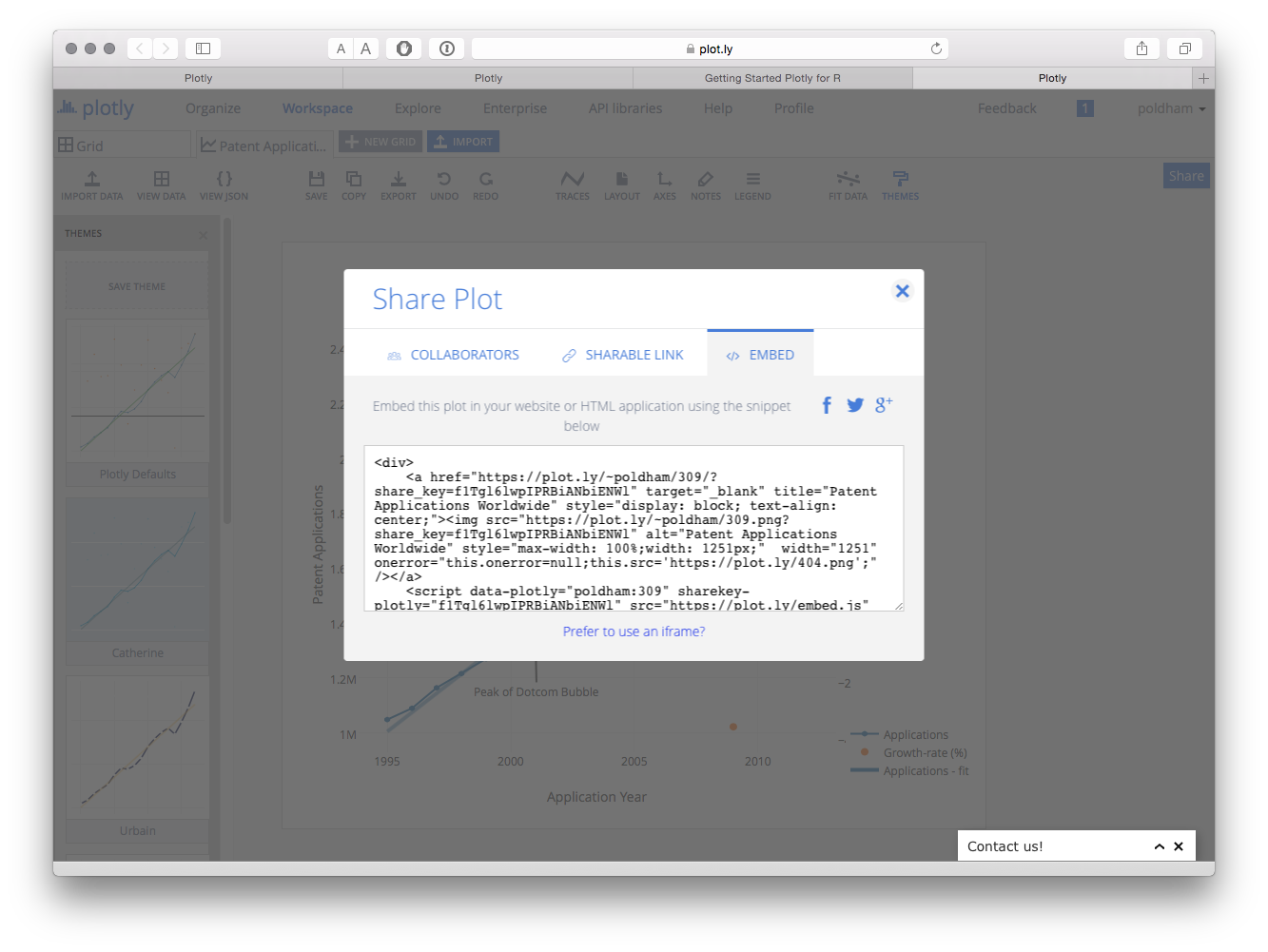
To save the plot we simply click Save. However, it is here that one of Plotly's major strengths becomes apparent. As soon as we save the plot we can also invite others by email, we can create a public or private shareable link. For the collaborators, they must have a Plotly account already for this to work.



The next option is to share a link. Note here that the default is to share a private link. To change that select the lock icon. The private link is particularly well suited for patent professionals.



You could also grab an embed code to embed the plot in a web page



Alternatively, surprise your friends and relatives by posting the plot on facebook or share with a wider audience on Twitter.

In this example we have focused on developing a very simple plot using plotly. In practice there are a wide range of possible plotting options with a range of tutorials provided [here](http://help.plot.ly/)

## Working with Plotly in R

We are following the instructions for setting up Plotly in R [here](https://plot.ly/r/getting-started/). We will be using [RStudio](https://www.rstudio.com/) for this experiment. Download RStudio for your operating system [here](https://www.rstudio.com/products/RStudio/). For Python try these [installation instructions](https://plot.ly/python/getting-started/) to get started.

In RStudio first we need to install or load the devtools package.

install.packages("devtools")

Then load the library.

library(devtools)

The we install the plotly package.

devtools::install\_github("ropensci/plotly")

When we load the library it will load other required packages, note that you may need to install some of these packages if you don't have them already. Use install.packages("ggplot2") and so on in the Console if this happens and then load the libraries.

library(plotly) ## loads Plotly and the additional packages it needs.

Loading required package: RCurl Loading required package: bitops Loading required package: RJSONIO Loading required package: ggplot2

We now need to set our credentials for the API. Follow this [link](https://plot.ly/settings/api) to obtain your API key (when logged in to Plotly). Note also that you can obtain a streaming API token on the same page. Streaming will update a graphic from inside RStudio.

When you have obtained your token then use the following command to register.

set\_credentials\_file("your username", "your API key")

To start plotting we will create an empty plotly object. We will call it pizza plot but you can call it what you like.

pizzaplot <- plotly()

If we were to use class(pizzaplot) we would see that we have an object of "PlotlyClass".

Next we will use a quick plot or qplot from ggplot2. First we will load the pizza patents by country and year dataset from the Github repository using readr. First you may need to install the readr package and/or lead the library.

install.packages("readr")

Load the library.

library(readr)

Now read in the dataset from the Github respository (if you have it downloaded already you could load by inserting the path to your local file inside the quotes).

pcy <- read\_csv("https://github.com/poldham/opensource-patent-analytics/raw/master/2\_datasets/pizza\_medium\_clean/pcy.csv")

We will now make a quick plot using qplot from ggplot2. The data is for trends in patent documents mentioning pizza from WIPO Patentscope. We have set a limit to the data for 1970 to 2012 to edit out sparse data and remove the data cliff for recent years.

pizza <- qplot(pubyear, n, data = pcy, geom = "line", colour = pubcountry, xlim = c(1970, 2012))

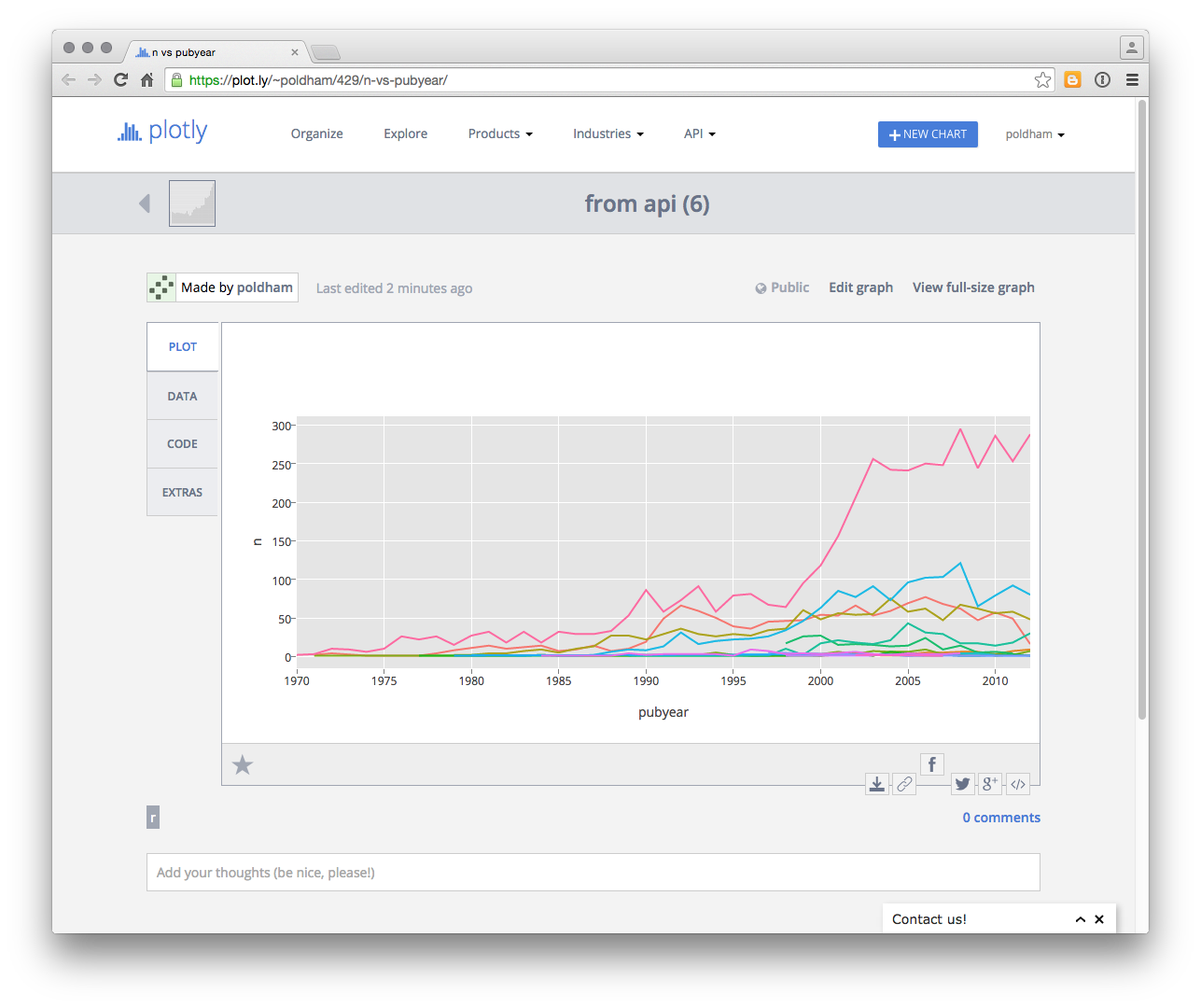
To create the plot in plotly we need to run the following.

response <- pizzaplot$ggplotly(pizza)

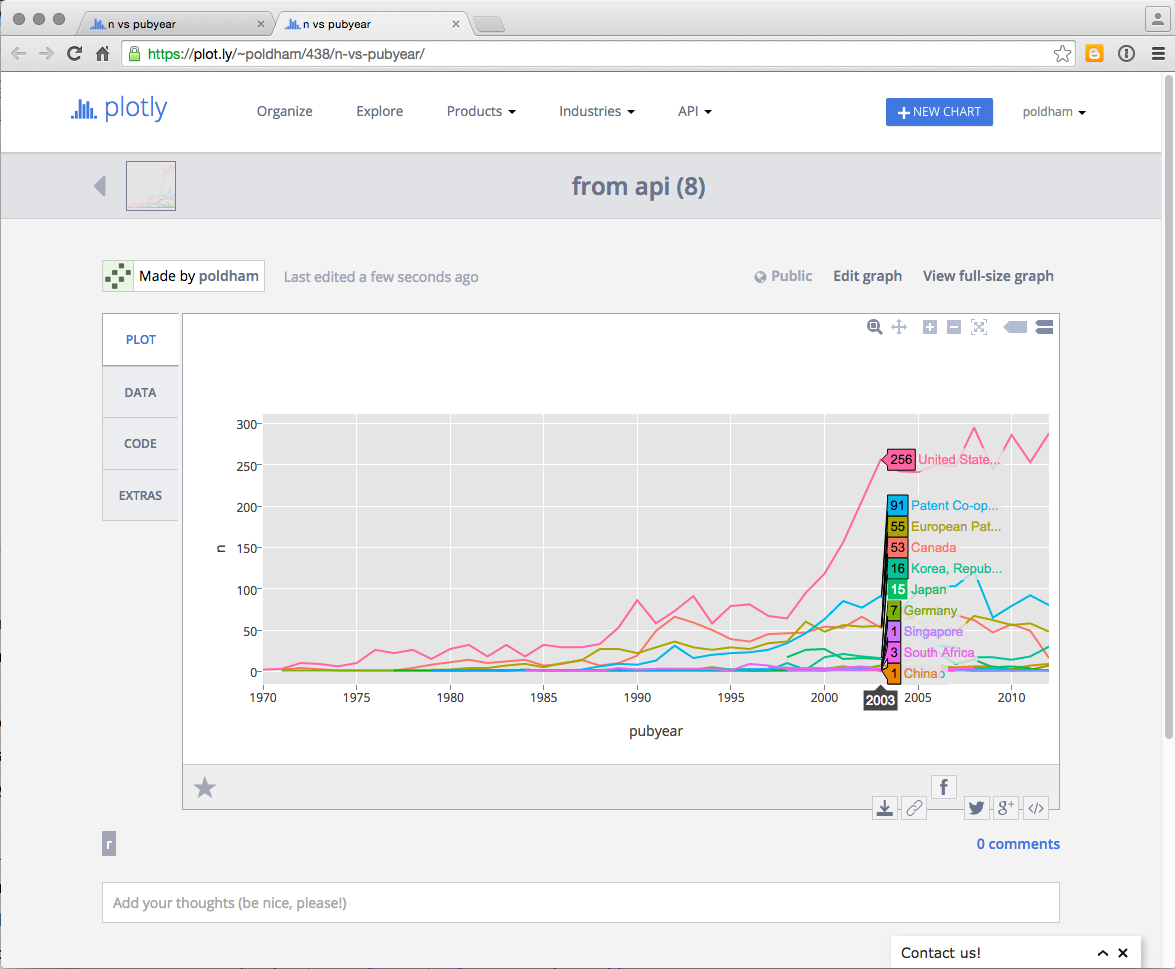
Let's go and visit it online.

response$url

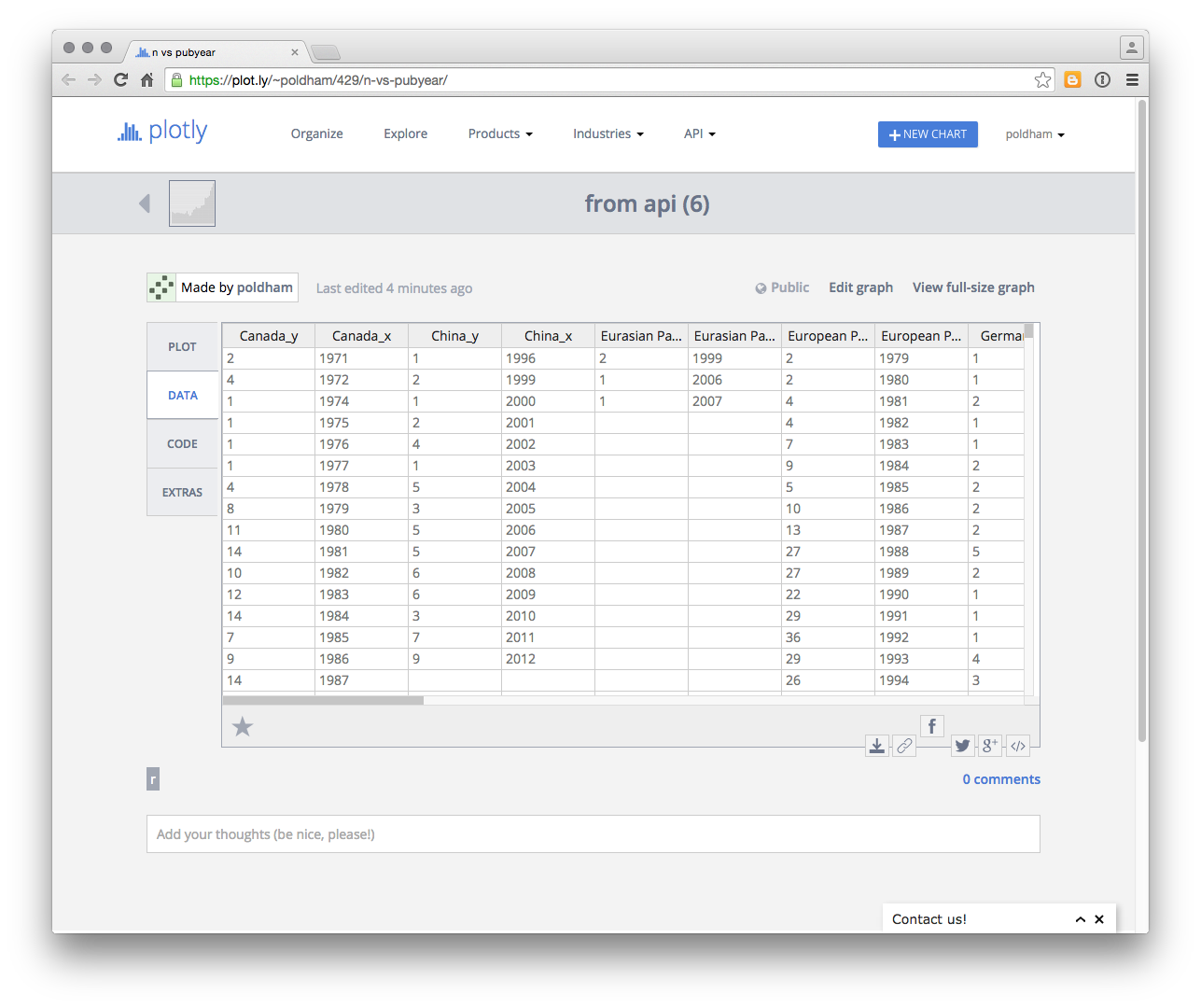
You will now see an online plot that should look like this.



If we hover over the data points we will also see the data appear by country (to add a legend, edit graph and then Legends)



Note here that the data table is also provided under the Data tab.



We can also share the graph via social media, download the data, or edit the graph. Note that the default setting for a graph sent via the API appears to be public (with no obvious way to change that).

It is here that Plotly's potential importance as a tool for sharing data and graphics becomes apparent.

## Round Up

In this article we have provided a brief introduction to Plotly to help you get started with using this tool for patent analytics. Plotly provides visually appealing and interactive graphics that can readily be shared with colleagues, pasted into websites and shared publicly. The availability of APIs is also a key feature of Plotly for those working in Python, R or other programmatic environments.

However, Plotly can also be confusing. For example, we found it hard to understand why particular datasets would not upload correctly (when they can easily be read in Tableau). We also found it hard to understand the format that the data needed to be in to plot correctly. If we examine the data table above it is clear that Plotly has converted each country in the underlying data (which is in long format) into individual x and y axes. We experienced significant problems with making datasets that work fine in Tableau work in Plotly. So, Plotly can be somewhat frustrating although it has very considerable potential for sharing appealing graphics. As is so often the case, this will also involve significant investments in time to understand the way Plotly works and in particular the format for the data that works best with Plotly.

In this article we have only touched on the potential of Plotly. Other kinds of plots that are well worth exploring include Bubble maps, contour maps and heat maps. To experiment for yourself try the Plotly [tutorials](http://help.plot.ly/tutorials/).