Plotalot for Joomla

USER GUIDE

Version 7.05
24 April 2024

CONTENTS

Plotalot makes it easy to create live charts and place them on your site. You must be able to write the SQL queries to retrieve the data you want to plot, but if you can do that, Plotalot will take the data and use the Google Charts API to produce charts or tables of your data.

You can create menu items that display charts on your website, or you can use the Plotalot Plugin to include as many charts as you like in any article, simply by writing {plotalot id="xxx"} anywhere in the article. The Plotalot plugin will replace any number of {plotalot} tags with an image or table, as defined by the chart definition.

Some of Plotalot's features are:

- Numerous chart types, with numerous options
- Charts can dynamically responsive, for example changing size when a device is rotated
- Charts can dynamically refresh without re-drawing the entire page.
- Simple tables, Google Charts tables, or single items of text
- Titles and axis labels can be simple text or SQL queries
- Axis limits can be automatic or specified by numbers or SQL queries
- Query the local site database or any supported external database
- No libraries to install, just a component and a plugin
- Show charts on your site front end using a simple menu item, or using the Plotalot article plugin
- Self contained charting class can be built in to your own components (front end or back end)
- The Plotalot Plugin can cache charts to reduce database load and improve performance
- Charts can be fully responsive
- Charts can automatically refresh using Ajax, i.e. without redrawing the whole page

LICENCE

Plotalot is open source software licensed under the GNU General Public License Version 2, the same licence that Joomla itself uses. The GNU General Public License is a Free Software license. For more details please see http://www.gnu.org/licenses/old-licenses/gpl-2.0.html

This document is copyright Les Arbres Design. It may not be reproduced or redistributed without the permission of the copyright holder.

SQL

Plotalot is easy to use, but there is no getting away from the fact that you need to use SQL (Structured Query Language). This guide includes a basic introduction to SQL and contains many examples that should help you, but learning SQL does take time. SQL can become very complex, although luckily, here, we are only concerned with "Select" queries. SQL is a huge subject and there are many references on the web. The ultimate reference guide for MySql is the MySql Documentation at http://dev.mysql.com/doc/

Please note that we do not usually provide support for SQL problems, although we are always interested to gather examples to add to this guide.

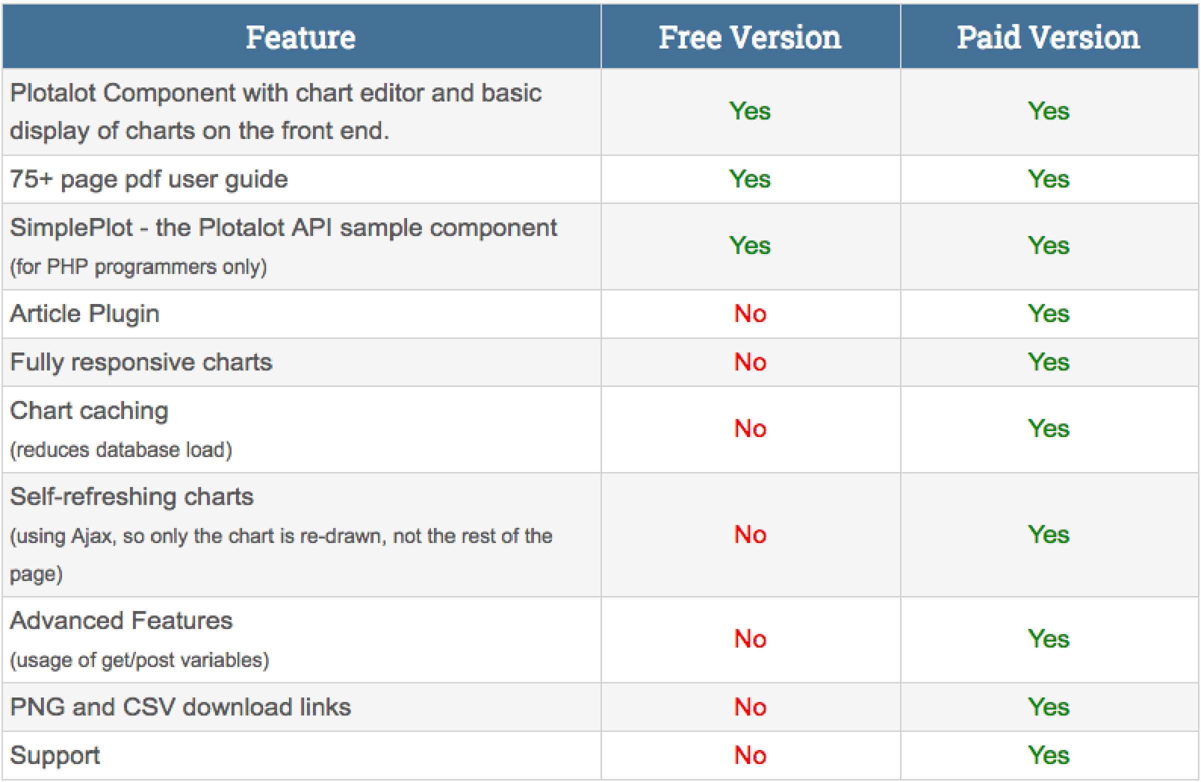
THE FREE AND PAID VERSIONS OF PLOTALOT

The Plotalot Component is completely free and fully functional. The component lets you create and edit charts and display them on the front end of your site using a menu item.

The paid version of Plotalot includes the Plotalot Plugin. The main function of the plugin is to merge charts into a Joomla article. It also adds some advanced features, listed below.

We suggest that you experiment with the free Plotalot Component first, and only buy the paid version once you are sure that you can create the charts you need.

This table summarises the differences between the free and the paid versions.



Feature	Free Version	Paid Version
Plotalot Component with chart editor and basic display of charts on the front end.	Yes	Yes
75+ page pdf user guide	Yes	Yes
SimplePlot - the Plotalot API sample component (for PHP programmers only)	Yes	Yes
Article Plugin	No	Yes
Fully responsive charts	No	Yes
Chart caching (reduces database load)	No	Yes
Self-refreshing charts (using Ajax, so only the chart is re-drawn, not the rest of the page)	No	Yes
Advanced Features (usage of get/post variables)	No	Yes
PNG and CSV download links	No	Yes
Support	No	Yes

If you decide to buy the paid version, you still use the same free Plotalot Component to create and edit your charts. But with the paid version you would normally use the plugin to merge your charts into Joomla articles.

If you use the free version, you only need to install the Plotalot Component.

If you decide to use the paid version, you need to install the Plotalot Component and the Plotalot Plugin.

PLUGIN VERSION DEPENDENCIES

We do our best not to break compatibility between the Component and the Plugin but unfortunately this does sometimes happen with major new features.

For example, at version 6 we changed to Google's new way of loading the chart library code. Unfortunately this required simultaneous changes in both the component and plugin, with the result that older versions of the plugin no longer work with version 6 and later of the component.

PLOTALOT DATABASE SUPPORT

Plotalot can retrieve chart data from external databases using a range of database drivers provided by Joomla. Drivers provided by Joomla include MySql, Oracle, Postgre, MS Sql, and Sqlite. These drivers in turn depend on PHP extensions and/or other software that may or may not be available on your server platform.

The database configuration interface in Plotalot shows a list of database drivers installed and attempts to determine their support status on your server.

The Joomla site that Plotalot is installed on must be running on a MySql database.

TECHNICAL REQUIREMENTS

The Plotalot component and plugin work in Joomla 3.7.0 and above, and Joomla 4.

The Joomla site that Plotalot is installed on must be running on a MySql database.

Plotalot can create charts from external databases other than MySql, but its Joomla database must be MySql.

INSTALLING

Install the Plotalot Component using the Joomla Extension Manager in the usual way. If you have bought the paid version, also install the Plotalot Plugin.

In the admin interface, you will find a new component "Plotalot", where you can create and edit charts and tables. You will also now be able to create menu items of type “Plotalot, Simple List of Charts”. See Showing Charts Using a Menu Item for more details.

To install the Plotalot plugin, use the Joomla Extension Manager in the usual way. Check that it is enabled in the Joomla Plugin Manager. Then see Adding Charts to Articles - The Plotalot Plugin for more details.

UPGRADING

You can always upgrade from any version to any newer version without uninstalling. You will not lose your data. Just install the new version in the usual way.

You can, if you wish, uninstall the old version and then install the new version. You will never lose your Plotalot data by uninstalling the component. However, you may need to edit and re-save any Plotalot menu items.

UNINSTALLING

To uninstall Plotalot, use Extension Manager: Manage in the usual way. All the files that were installed will be removed, but the database table is not deleted. If you want to remove all trace of Plotalot permanently, you can manually drop the three xxx_plotalot tables using a utility such as PhpMyAdmin.

SAMPLE CHARTS

A number of sample charts are installed with Plotalot. You can safely delete them, but they will be re-installed every time you re-install Plotalot. You can modify the sample charts but you should not use them as live charts because they are permanently flagged as samples and might, in a future version, be modified by a re-installation of Plotalot.

UPDATING THE PLOTALOT PLUGIN

The Plotalot Plugin is not free and our website requires a valid purchase ID for downloads of paid software.

Your Purchase ID

Your 32 character purchase ID is provided in the purchase confirmation email. If you lose the email you can re-send it by entering the purchase email address on this page of our website:

https://www.lesarbresdesign.info/support-information/download-link-recovery

*Please note that your purchase ID is **not** your PayPal (or Stripe) transaction ID. PayPal assigns separate, private, transaction ID's to its merchant and buyer transactions.*

Save Your Purchase ID on the About Page

You can update the Plotalot plugin using the Joomla Extensions Update page, but for this to work, Joomla has to include your purchase ID on the download request to our website. To enable this you must enter and save your purchase ID on the About page of Plotalot. The About page then also shows your subscription status. Updates will fail if you have not entered your purchase ID on the About page, or if your subscription has expired.

Joomla caches update information, so if you enter your purchase ID on the About page and then immediately try to update, the update might still fail. Use the "Check For Updates" button to refresh the update information.

Manual Downloads

The purchase confirmation email includes a link to your personal download page, where you can always download the latest software to your local computer. We recommend that you always keep a copy of the latest version in its installable form, together with its matching user guide version, in case you need to re-build your website. The URL of your personal download page looks like this:

https://www.lesarbresdesign.info/my-page?tid=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Where xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx is your purchase ID.

Joomla Update Caching

By default Joomla checks for updates every six hours but you can change this by changing the Update Caching parameter in Installer Options. If you set Update Caching to zero, Joomla checks for updates every time you load the Joomla admin control panel. This can result in high levels of traffic to our server, so if you do this, Plotalot disables its update checking. Only the update checking for Plotalot is disabled. Other extensions are not affected.

SQL For Beginners

Since SQL is so central to Plotalot, we need to discuss it before we proceed to build any charts. If you already know SQL, you can skip this section.

SQL is a database query language. It enables humans and programs to create, read, update, and delete data in databases in a (sort of) standardised language, the Structured Query Language. An instruction to a SQL database is known as a query, and queries can range from simple ones that almost anyone can understand, to complex monstrosities that almost no-one can understand. Fortunately, here we are concerned only with queries that retrieve data - "Select" queries. With a little knowledge, you can achieve a great deal, and since Plotalot only allows select queries, you can't do any damage, so let's have a go!

In this guide, we show SQL verbs and functions in UPPER CASE, with column and table names in lower case. We often omit `backticks` and "quotes", although please see the comments about this later.

SOME TERMINOLOGY

Databases are collections of *tables*. Tables are a bit like files, although they work very differently. You might prefer to think of them as analogous to the worksheets in a spreadsheet. Each table has a name.

A Joomla database has numerous tables. Extensions often add their own tables. A few of the core Joomla tables are:

xxx_content
xxx_content_frontpage
xxx_content_rating
xxx_extensions
…

We can take a good guess at what most of these tables are used for. The xxx_ here is the database table name prefix configured when the site is installed.

Each table can contain many *rows*, rather like the rows on a spreadsheet, and each row can contain many *columns*. Each column has a name, which describes the data item it holds. Rows don't have names, although in Joomla they usually have an id number that uniquely identifies them.

Here's an extract from the xxx_content table:

id	title	alias	...	...	hits
1	Welcome	welcome			496244
2	Article 1	article1			39945
3	Article 2	article2			467777

We don't need to know what most of these columns are for to notice that two of them look as though they could be useful to us: the "Title" and "Hits" columns. We will make use of these in our first chart. Columns are sometimes referred to as *fields*. The *columns* and their names are defined by the author of the software that creates or uses the table. The *rows* are usually added at runtime by using the software.

We usually want to ask the database to get certain rows for us based on the data they contain - the row for a particular order number, for example, or perhaps all the rows for all the orders on a particular date. The data that comes back from such a request is called a *result set*, or a *dataset*. In charting, we also refer to it as a *data series*.

TOOLS OF THE TRADE

You could just enter queries straight into Plotalot. You will see the results, or the errors in your SQL, but it would be a frustrating environment in which to learn SQL. More typically you will use a tool like phpMyAdmin. Most web hosts give you access to this tool, and it's a much easier place to develop queries than Plotalot. Once you are in phpMyAdmin, select your database, and click the SQL tab to bring up the SQL query utility. You're ready to go!

SIMPLE QUERIES

The only SQL queries that Plotalot allows are of the form "Get Me Something from Somewhere". In SQL that translates to:

```
SELECT field_name(s) FROM table_name(s)
```

These queries extract data from a database but do not allow any changes to be made to the database. You can add many functions to format the results the way you want, select the rows that come back, sort them, and restrict the number of rows returned. One of the simplest ways to query a database is to ask for all the columns of all the rows in a particular table. Try this:

```
SELECT * FROM xxx_content
```

Note that we use UPPER CASE to denote SQL language constructs, and lower case to denote names of things. You don't have to type SQL that way, we just do it here to help you see what is SQL and what are data names.

The query above brought back all the columns in the table, including many that we aren't very interested in. Try restricting the number of columns you bring back by naming the ones you want.

```
SELECT title, hits FROM xxx_content
```

Order By

SQL can sort rows into any order you want. For our "Most Popular" chart, we will want the content articles with the most hits, so we need to sort them. Try this:

```
SELECT title, hits FROM xxx_content ORDER BY hits
```

Well, it's close, but we got the least popular ones first - ascending order is the default. We can reverse the order by adding the DESCending clause:

```
SELECT title, hits FROM xxx_content ORDER BY hits DESC
```

Limit

We're almost there, but at the moment we are still bringing back all the rows in the whole table, and that would make a real mess of a pie chart. We really only want the ten most popular articles. In MySql, we can use the LIMIT clause:

```
SELECT title, hits FROM xxx_content ORDER BY hits DESC LIMIT 10
```

That's it. Now we have a list of the ten most popular articles on the site, starting with the most popular at the top. This is exactly the query we need for our first chart!

Where

Another, more powerful, way to restrict the dataset, is to tell the database more about the rows we really want. Let's say we want all the articles that have been read more than 10,000 times:

```
      SELECT title, hits FROM xxx_content WHERE hits > 10000
```

Or how about the ten most popular articles published since a specific date:

```
      SELECT * FROM xxx_content WHERE created > "2020-01-01"
            ORDER BY hits DESC LIMIT 10
```

Or between two dates:

```
SELECT * FROM xxx_content WHERE created BETWEEN "2020-01-01" AND "2020-12-31"
      ORDER BY hits DESC LIMIT 10
```

MORE USEFUL QUERIES

That was the gentle introduction. It's easy to get started with SQL, but real life queries are not always so easy. You will need to practice, Google, and practice some more before you become really confident.

Very simple queries like those above can only return data in its raw form, unchanged from the way it is stored in the database. Very often, you need to manipulate the data in some way.

Functions

Try this:

```
      SELECT NOW()
```

It returns the current date and time. The () notation indicates a function that performs some action and returns a value. In this case there was no input to the function, only an output. Functions can have inputs too.

```
      SELECT MONTHNAME(NOW())
```

This time we passed the output of the NOW function into the input of the MONTHNAME function, which returns the name of the month as its output.

Date and Time Functions

In databases, dates and times are usually held in a special format called a DateTime. We don't need to know how that format works inside the database, just that it is always converted to a human readable date on its way back to us. By default, the output format is "2020-07-28 15:10:37". That might not always be what we want, so SQL has a range of functions to convert dates and times to other formats. Here are some of them:

DATE_FORMAT()	Format a date however you want
CURRENT_DATE()	Get the current date
DATE_ADD()	Add an interval to a date
DATE_SUB()	Subtract an interval from a date
UNIX_TIMESTAMP()	Convert a DateTime to a Unix timestamp
FROM_UNIXTIME()	Convert a Unix timestamp to a SQL date
DATE()	Extract just the date part of a DateTime

There are many more, but these will get you started. Let's have a look at a few examples.

Example 1

Let's say we have a table with columns "sdate" and "total_sales", where sdate is a "DateTime" column and num_sales is an "int" column, and we want to plot a graph of sales over time. A Plotalot line chart requires dates to be converted to Unix Timestamp format (the number of seconds since the 1st of January 1970), so to plot our chart we would need to use the query:

```
SELECT UNIX_TIMESTAMP(sdate), total_sales FROM my_table
```

Example 2

Dates and times are usually stored in databases using the DateTime column type, but not always. You might have a table where the dates are stored in Unix Time format. That would be perfect for drawing a Plotalot line chart, but what if we want to show the data as a readable table? You would need to convert the Unix dates to a readable format.

The solution is to first convert the Unix dates to SQL dates using FROM_UNIXTIME, and then send the output of FROM_UNIXTIME into the DATE_FORMAT function:

```
SELECT DATE_FORMAT(FROM_UNIXTIME(payment_date), "%d-%m-%Y") AS `Payment Date`
```

It's common to have to nest functions like this. You can lookup the syntax for the DATE_FORMAT function to get the date in the format you want. Google "mysql date_format" to find the full list.

In this example we also used the AS keyword to rename the output column to "Payment Date". If we didn't do that the table column title would be "DATE_FORMAT(FROM_UNIXTIME ... ".

Example 3

By default, chart axes are automatically adjusted to span the data being charted, but sometimes we want to force the axes to be larger than the dataset. Plotalot lets you use SQL queries to override the chart axis limits. So in a chart where the X axis represents dates, we could, for example, force the X axis to run from the beginning of a particular year to its end:

```
X axis start:    SELECT UNIX_TIMESTAMP(DATE('2020-01-01'))
X axis end:      SELECT UNIX_TIMESTAMP(DATE('2020-12-31'))
```

Or for the current year:

```
X axis start:    SELECT UNIX_TIMESTAMP(DATE_FORMAT(NOW(),'%Y-01-01'))
X axis end:      SELECT UNIX_TIMESTAMP(DATE_FORMAT(NOW(),'%Y-12-31'))
```

String Functions

SQL can also manipulate strings. There are many functions, but a few of the most useful are:

CONCAT()	Concatenate (join) strings together
FORMAT()	Return a number as a string, formatted to a specified number of decimal places
SUBSTR()	Return part of a string

Example 4

Create a title for a chart like "Hits Today (Wednesday)"

```
SELECT CONCAT("Hits Today (", DATE_FORMAT(CURDATE(),"%W"), ")")
```

Example 5

Truncate a text column to ten characters:

```
      SELECT SUBSTR(title,1,10)
```

Example 6

Format a numeric column as a string:

```
      SELECT FORMAT(price,2)
```

ADVANCED QUERIES

We have shown you how easy SQL can be, and hopefully we have shown that with a little patience you can achieve quite a lot. We will now briefly show you how powerful SQL can be. We won't attempt to explain how all these examples work in detail. If you are new to SQL it will be a while before you can write queries like these, but we hope they will give you an idea of what is possible.

Example 7

We have a Joomla extension that creates a row in a table for each sale we make. We want to build a bar chart of sales by date. As you will see later, a bar chart requires a text column for the bar labels and a numeric column for the bar height. The dates in our table are stored as Unix Date-Time. We need SQL to calculate the total amount for an item, for each date within a date range. Here's the solution:

```
SELECT DATE_FORMAT(DATE(FROM_UNIXTIME(payment_date)),"%d") AS pay_date,
      SUM(price) AS total_price,
      DATE(FROM_UNIXTIME(payment_date)) AS sql_date
      FROM #__my_sales
      WHERE item_number = 1
      AND (DATE(FROM_UNIXTIME(payment_date))
            BETWEEN DATE("2020-01-01") AND DATE ("2020-03-01"))
      GROUP BY sql_date
```

The GROUP BY clause here takes the rows for each date and lets you apply "aggregate" functions like SUM to them. Here, we used SUM to get the total sales for each date. The extra column (sql_date) is created purely for use by the GROUP BY clause. Plotalot will use the first two columns to create the chart, and ignore any extra columns.

Example 8

We want the article hits pie chart to include a slice for all the "Other" articles not in the top ten:

```
SELECT * FROM
      ((SELECT title, hits
            FROM #__content ORDER BY hits DESC LIMIT 10) AS temp1)
UNION
      (SELECT 'Other' AS title, SUM(hits)
            FROM (SELECT title, hits FROM #__content ORDER BY hits DESC
                  LIMIT 10,18446744073709551615) AS temp2)
```

The first sub-query gets the top ten articles as before. UNION tells SQL to add two result sets together. The second sub-query gets the sum of the hits for the articles not in the top ten. Try it!

Example 9

Here's a table containing only one column (called "data"), with the data as name=value pairs. The first four rows of the table look like this:

row 1:	x_value=33||y_value=41.5||z_value=39.2
row 2:	x_value=34||y_value=42.3||z_value=44.5
row 3:	x_value=35||y_value=47||z_value=31
row 4:	x_value=36||y_value=48.9||z_value=53.6

We need a line graph of the x values against the y values. Here's the solution:

```
SELECT CAST(SUBSTRING_INDEX(SUBSTRING_INDEX(data, 'x_value=', -1), '||', 1)
            AS SIGNED INTEGER) AS x,
      CAST(SUBSTRING_INDEX(SUBSTRING_INDEX(data, 'y_value=', -1), '||', 1)
            AS SIGNED INTEGER) AS y
      FROM #__name_value_data ORDER BY `y`
```

Which returns this result set:

x	**y**
33	41
34	42
35	47
36	48

SQL For Plotalot

This section contains general information about Plotalot SQL queries that all users need to be aware of.

DATABASE TABLE PREFIXES

Joomla database table names begin with a prefix, which is chosen at site installation time and stored in the configuration.php file, as a first line of defence against SQL attacks. In Plotalot queries, you can use the table names as they actually exist in the database with the configured site prefix, but to make your charts more portable it's better to use the universal prefix "#__"

```
SELECT `title`, `hits` FROM `#__content` ORDER BY `hits` DESC LIMIT 10
```

The "#__" is dynamically replaced with the site prefix, so if the site prefix is "cdxz_", this query resolves to:

```
SELECT `title`, `hits` FROM `cdxz_content` ORDER BY `hits` DESC LIMIT 10
```

MULTIPLE COLUMNS WITH THE SAME NAME

Some SQL tools allow you to work with queries like this:

```
SELECT 'data_1' AS `column_1`, 'data_2' AS `column_1`;
```

This won't work in Plotalot. All columns must have unique names. In Plotalot, if two or more columns of a SQL result have the same name, the last column takes precedence and overwrites data from a previous column with the same name.

UNIONS IN THE SAMPLE CHARTS

Many of the Plotalot sample charts use the SQL Union operator to return multiple rows of fixed data:

```
SELECT 1 AS X, 3 AS Y
UNION SELECT 2, 0
UNION SELECT 3, 3
```

This is just a convenient way of providing data for the sample charts without needing a sample data table. The SQL statement above returns three rows, each with two columns. Plotalot doesn't care what SQL you use, as long as it starts with a "Select".

You wouldn't normally write queries like this, but if you do, make sure to use the "AS" clause for the first row to name the columns. This query will only return one column in Plotalot, because both columns have the name '1':

```
SELECT 1, 1
UNION SELECT 2, 0
```

COLUMN DATA TYPES

Each chart type has specific requirements for its data columns, either text or numeric. To avoid Javascript errors, Plotalot does not allow charts to be created with the wrong column types.

Plotalot determines the data type of a column by examining the data returned. It does not look at the database column definitions. Text columns can consist of text and/or numbers. Numeric columns must only contain numbers.

The query requirements for each column are listed with the detailed descriptions of each chart type, later in this guide.

NULLS

Some database queries can return null values, especially where joins are used. By default, when Plotalot finds nulls in SQL results, it changes them to an empty string for string columns, or zero for numeric columns. It also shows a warning in the chart editor. Warning are not shown on the front end.

Plotalot determines the column types as described above before fixing nulls.

Sometimes you might want null values to be included in your chart data as actual null values, for example if you use the Google Charts InterpolateNulls option to calculate missing data points, or if you want to see actual gaps in your charts. In such cases you can set the Plotalot Component Option "Fix nulls" to "No".

If neither of these options is right for you, you can use the SQL IFNULL( ) function to return a specific value in place of a null value.

DATES AND TIMES

For some Plotalot charts, dates and times must be retrieved as Unix timestamps (the number of seconds since the 1st of January 1970). Dates and times are usually stored in databases as date, datetime or timestamp column types, so you usually need to convert them in your queries. For example:

```
SELECT UNIX_TIMESTAMP(`purchase_date`), `quantity` FROM `my_table`
```

The query requirements for each column are listed with the detailed descriptions of each chart type, later in this guide.

ADDING A HORIZONTAL LINE

In the previous example you might want to add a fixed horizontal line on the chart to represent your target quantity. That's easy. Just add an extra plot with two points, one at the minimum date and one at the maximum date, both with the same value:

```
SELECT UNIX_TIMESTAMP(MIN(`Date`)),120 FROM `my_table`
UNION
SELECT UNIX_TIMESTAMP(MAX(`Date`)),120 FROM `my_table`;
```

For a horizontal line at the average quantity value, try this:

```
SELECT UNIX_TIMESTAMP(MIN(`Date`)),AVG(`quantity`) FROM `my_table`
UNION
SELECT UNIX_TIMESTAMP(MAX(`Date`)),AVG(`quantity`) FROM `my_table`;
```

TIMEZONES

Plotalot does not do any timezone conversions. It's usually best to store times as UTC and convert to the required timezone when you retrieve the data. For example:

```
      CONVERT_TZ(`my_date_field`, 'UTC', 'Europe/Paris')
```

or:

```
      CONVERT_TZ(`my_date_field`, '+00:00', '+01:00')
```

QUOTES

In MySql queries, table and column names should be quoted with `backticks`, and text values should be quoted with 'single' or "double" quotes. MySql allows you to be lazy and omit the quotes, as we sometimes do in this guide. But you can't always omit them. Sometimes, queries won't work without them. Let's say we have a table with column called "table". This query will return a syntax error because 'table' is a reserved word:

```
      SELECT table, name FROM my_table
```

In this case you must write:

```
      SELECT `table`, `name` FROM `my_table`
```

Sometimes, a query might appear to work, but does not return the results you might be expecting. Let's say we have a table of annual sales results with columns "month_txt", "2017", "2018", "2019", "2020". This query will execute successfully:

```
      SELECT month_txt, 2020 FROM my_table
```

But it won't give you the values of the 2020 column. It will return the literal numeric value 2020…

```
      January, 2020
      February, 2020
      March, 2020
```

To clarify that 2020 is a column name, you must use the backticks:

```
      SELECT `month_txt`, `2020` FROM `my_table`
```

It's good practice to always use backticks for table and column names, and normal quotes for text values.

CSV FILE DATA

Plotalot can retrieve data from a CSV file instead of a database. You can specify a query like this:

```
      FILE tmp/plotalot_data.csv
```

The query must start with the word "FILE" (case insensitive) followed by a single space, followed by a file path. If the file path starts with a forward slash ("/") the path is assumed to be absolute. If the file path does not start with a forward slash it is assumed to be relative to the site root directory. The file can have any name, it does not have to have the .csv extension. File names may contain spaces but make sure there are no spaces or other invisible characters after the file name.

The file must be comma or tab delimited, and is assumed to have a heading row with at least the number of columns required by the chart. All other rows must have the same number of columns as the heading row.

Files may have more columns than required. Additional columns to the right are ignored, but the leftmost column is always treated as the first column.

A simple example of a CSV file would be like this:

```
Column 1, Column 2
1, 2
2, 3
```

Plotalot line charts require dates to be in Unix Timestamp format. In Excell you can convert readable dates to Unix Timestamps with the formula: =(A1-DATE(1970,1,1))*86400. For example, if you have a spreadsheet of dates and values, you can insert a new column using the formula above to calculate the Unix timestamp values:

Date	Unix date	Value
01/01/23	1672531200	43
02/01/23	1672617600	42
03/01/23	1672704000	42.5
04/01/23	1672790400	33.5
05/01/23	1672876800	31
06/01/23	1672963200	27

You would then need to copy columns B and C to another sheet and save as CSV to obtain a file like this:

```
Unix date, Value
1672531200, 43
1672617600, 42
1672704000, 42.5
1672790400, 33.5
1672876800, 31
1672963200, 27
```

HTTP DATA

Plotalot can retrieve data from an HTTP source instead of a database. You can specify a query like this:

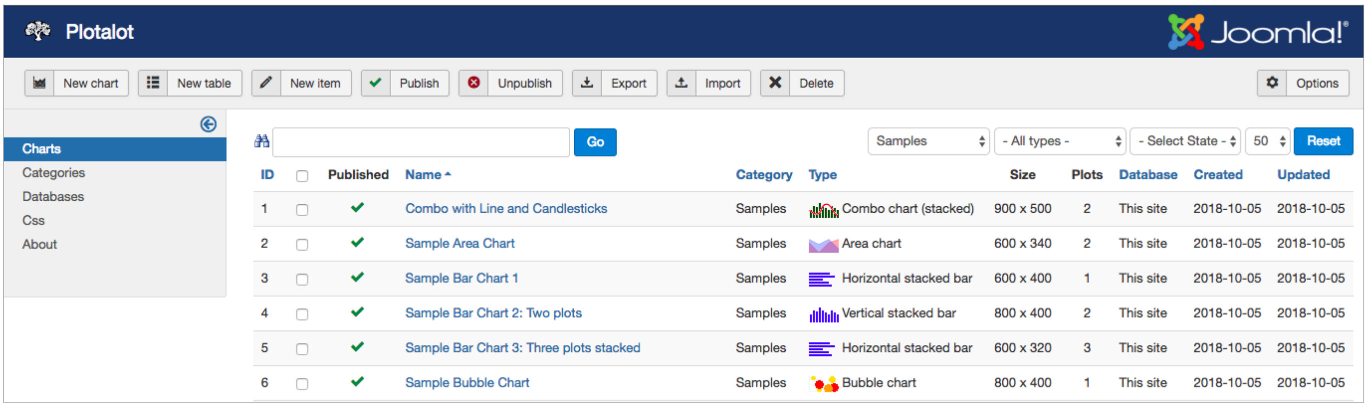
```
HTTP://sitename.com/plotalot_data.csv
```

The query must start with "HTTP" (case insensitive) and be a valid, freely accessible URL not requiring a login. The file can have any name, it does not have to have the .csv extension.

The file must be comma or tab delimited, and must have a heading row with at least the number of columns required by the chart. All other rows must have the same number of columns as the heading row.

When you select the Plotalot Component in the back end of Joomla, the first thing you see is the Chart List, where you can create new charts or edit existing charts.

THE CHART LIST



ID	☐	Published	Name ▲	Category	Type	Size	Plots	Database	Created	Updated
1	☐	✓	Combo with Line and Candlesticks	Samples	Combo chart (stacked)	900 x 500	2	This site	2018-10-05	2018-10-05
2	☐	✓	Sample Area Chart	Samples	Area chart	600 x 340	2	This site	2018-10-05	2018-10-05
3	☐	✓	Sample Bar Chart 1	Samples	Horizontal stacked bar	600 x 400	1	This site	2018-10-05	2018-10-05
4	☐	✓	Sample Bar Chart 2: Two plots	Samples	Vertical stacked bar	800 x 400	2	This site	2018-10-05	2018-10-05
5	☐	✓	Sample Bar Chart 3: Three plots stacked	Samples	Horizontal stacked bar	600 x 320	3	This site	2018-10-05	2018-10-05
6	☐	✓	Sample Bubble Chart	Samples	Bubble chart	800 x 400	1	This site	2018-10-05	2018-10-05

The filter buttons at the top of the chart list let you select the category, type, or published status of charts to include in the list view. You can search your charts using the search box, in which case all the other filters are ignored. The search function searches for any text anywhere in the charts, including the name, title, axis titles, extra options and SQL. Or you can enter a chart ID in the search box to quickly find that particular chart. The Reset button resets all the filters, search text and pagination, and refreshes the page.

You can delete charts but there is no undelete option. You can publish and unpublish charts. Unpublished charts are not shown on the front end of the site.

The Clear Cache button appears if any charts are currently being cached by the Plotalot plugin. Clicking the button deletes the cached charts. You might want to do that after you edit a chart that is cached so that your changes are immediately visible on the front end.

The Options button accesses the Plotalot configuration options, as discussed later.

Exporting and Importing Charts

The export and import buttons let you export selected charts in SQL format, and import SQL files containing Plotalot charts. It's an easy way to backup your charts, or move charts from one system to another.

To export charts, select the charts you want to export, and click Export. The export file is in SQL format and includes the selected charts as well as any categories they belong to, and any databases they use.

To import charts, click the Import button and select the file to import. When you click Save, the charts are imported, as well as any categories that are not already present, and any databases not already present.

When you export charts, they are exported *without* their unique chart ID, which means that if you re-import them back into the same system, they will be duplicated with a different chart ID. If you import them on a different system, they might well be imported with a different ID.

Categories and databases are exported *with* their unique ID, which means that if you import them to the same system, or another system that already has the same category or database ID's, the existing categories and databases will not be duplicated. They will only be created if they do not already exist.

Categories

Categories help you to organise your charts in the chart list. They have no other function. For example, there is no "category view" on the front end.

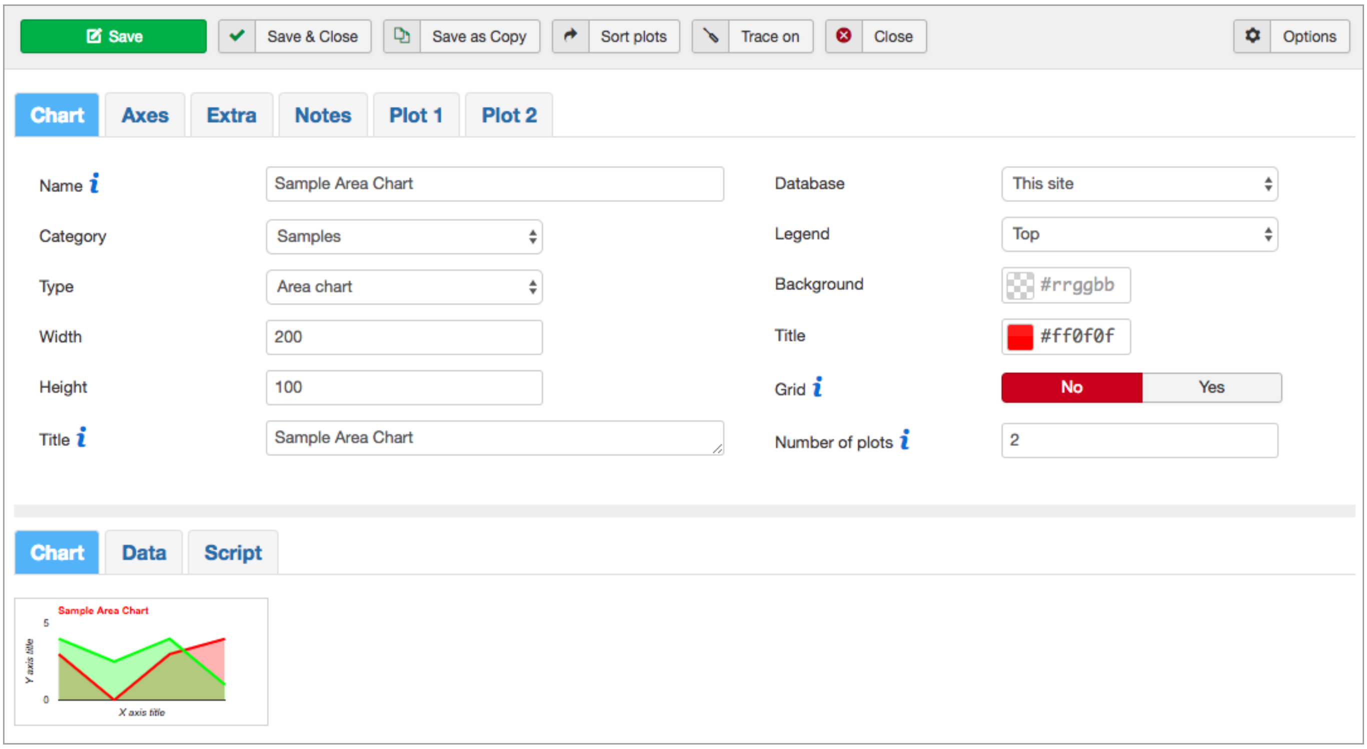
At installation, Plotalot installs some sample charts in the "Samples" category, and creates another category called "My Charts". You can't delete these two categories but you can rename them. "My Charts" is the default category for new charts. You can create as many other categories as you want.

You can rename charts and categories at any time. All internal linkages are by ID, not by name.

You can change or delete the sample charts. If you delete any of them, they are re-installed when you upgrade or re-install. You should not base your own charts on the sample charts as they always remain flagged as samples and might be over-written by future upgrades. Use the "Save as Copy" button to create your own copy of a sample chart. After you copy a chart, the copy is not flagged as a sample chart.

THE CHART EDITOR

Clicking on one of the charts in the Chart List takes you to the chart editor. The chart editor is organised as five or more tabs at the top of the page, and three or four tabs at the bottom.



Name: Sample Area Chart

Category: Samples

Type: Area chart

Width: 200

Height: 100

Title: Sample Area Chart

Database: This site

Legend: Top

Background: #rrggbb

Title: #ff0f0f

Grid: [✓ No] Yes

Number of plots: 2

This section is about charts. Tables and single items are discussed later.

The Chart Tab

This tab has options that affect the overall "look and feel" of the chart, including its size, type, background colour, and which database it retrieves data from.

The Axes Tab

This tab is used to customise the X and Y axes. You can use the fields here to label the axes, and to extend the range of the chart. If you leave the axis ranges blank, Plotalot will automatically assign the chart ranges to the limits of the data retrieved. There are some differences in the way these options work in different charts.

The Extra Tab

The extra tab is for advanced options that are described later. All charts work without anything being entered on this tab.

The Notes Tab

This is a single text area for your own use, for example to record changes to the chart over time, or notes about how it works. It is not part of the chart itself. It can contain a virtually unlimited amount of text.

The Plot Tabs

A plot defines the SQL query that retrieves one set of data values. All charts have at least one plot, and some chart types can have many plots. Each plot has a name, a colour, and some styling options that vary by chart type. For chart types that can have more than one plot, you can set the number of plots on the Chart tab.

The Chart Display Area

The display area shows any errors or warnings that apply to your chart, followed by three tabs. The Chart tab shows the chart itself, if it can be drawn. Each time you click Save, the chart is re-drawn using the options you have set. Developing a chart is an iterative process of setting options and clicking Save to see the results.

The Data tab shows the data that was returned from the database using the query or queries that you specified. By default only 20 rows of the data are shown because usually you are only interested in the formatting. A dropdown list at the right hand side of the data tab lets you select the number of rows to show.

The Script tab shows the Javascript that Plotalot generated for your chart. The Javascript tells the Google Charts API how to draw your chart. It's sometimes useful to see the Javascript when resolving problems or when working with Extra Options.

If Trace mode is enabled, the trace file is loaded into the Trace tab.

The Data, Script and Trace tabs are tools to help you develop your charts. They are not shown on the front end.

The Toolbar Buttons

Save saves the chart and returns to the chart editor.

Save and Close saves the chart and returns to the chart list.

Save as Copy saves the chart as a completely new chart and returns to the chart list. If you save a sample chart as a new chart, the new copy is not flagged as a sample chart.

Sort Plots saves the chart and returns to the chart editor with the plots sorted in ascending order by name. If you need to delete a plot, you can rename it to "zzz" and use the Sort Plots button to make it the last plot, then reduce the number of plots by one.

Trace On enables trace mode, which saves a lot of details about Plotalot's internal workings and calculations. Trace mode is mostly used for development and support of Plotalot, but is also useful to advanced users. When trace is enabled, the Trace On button changes to a Trace Off button, and trace output is included in the Trace tab in the display area. Note that the chart is NOT saved when you click the Trace On or Trace Off button. Save any changes first, using the Save button.

Cancel returns to the chart list without saving the chart.

Common Chart Options

Name

The chart name is only used for your reference. It does not appear in the chart itself.

Chart Type

The chart type defines how the data will be displayed. Each type of chart relies upon its data being in a particular format. For example line charts require two numeric columns.

Legend

The Legend shows which colour refers to which plot. You can set it to *None*, in which case it won't be displayed at all, or you can define where you want it to show. Not all of the legend options work with all chart types. Try them all to see which one works best for your chart.

Width and Height

The size of the chart, specified as *x* pixels wide by *y* pixels high. You can make charts responsive by setting the width to 0, but see the section Responsive Charts for details.

Database

To use a database other than the current Joomla site you must first create it using the "Databases" menu. You will then be able to select it here.

Title

The title displayed at the top of the chart. It can be a fixed string such as "My Chart" or it can be the result of a SQL query.

Title Colour

The title colour box is a colour picker. Click it to pick a colour or type in a hex colour code. If you leave the colour blank, the default is black.

Background

Choose or enter a colour for the chart background, or blank the field out to make the chart transparent. When the chart is transparent, you can place an image behind it to achieve gradient or stripe effects.

Grid

The grid can be on or off. When it is on, a faint grid is added to the chart background. The number of grid lines is controlled by the X Labels and Y Labels options.

Extra Options

If you can find your way around the Google Charts API Reference documentation, you can manually add additional options here. There are more details and examples in the Extra Options section. Be aware that Plotalot cannot validate the options you enter, so you can easily create errors here.

Extra Columns

A comma delimited list of additional column types. See the Extra Columns section for an example.

Extra Javascript

If you know Javascript there are many ways to enhance charts, for example adding "formatters" and interactive features. Please see the Extra Javascript section for more details.

CSV Link

If you have the Plotalot Plugin, you can add a link below the chart to download a .csv file containing the chart data. The link can include html, so it can be something like <button>Download CSV</button>. Leave this field blank for no CSV link. The CSV link is supported only by the plugin. It does not work in the component view.

PNG Link

If you have the Plotalot Plugin, you can add a link below the chart to download a .png image of the chart. The link can include html, so it can be something like <button>Download PNG</button>. Leave this field blank for no PNG link. The PNG link is supported only by the plugin. It does not work in the component view. Not all chart types support the conversion to .png images. If you see the error "getImageURI is not a function", the chart does not support this feature. This feature does not work in the IE and Edge browsers.

Locale

You can specify the Locale used by the Google Charts API, which affects the formatting of currencies, dates, and numbers. If you don't specify a Locale, the Locale specified in the Plotalot Component Options is used.

Number of Plots

The number of plots field determines the number of plot tabs shown in the chart editor. Set the number of plots you need and click Save. The editor screen will be re-displayed with the new number of plot tabs.

Some chart types can have multiple plots and some can only have one. The maximum is 20.

Design Pattern

For some charts you can select the Classic or Material version of the chart. Please see below for more information.

CLASSIC AND MATERIAL CHARTS

The Google Charts API now offers "Material design" versions of some of their charts. There's more information here:

https://developers.google.com/chart/interactive/docs/gallery/linechart#creating-material-line-charts

Plotalot supports Material charts with the "Design Pattern" option, which allows you to select the Classic or Material options. Currently, only line and bar charts are supported in the material design pattern.

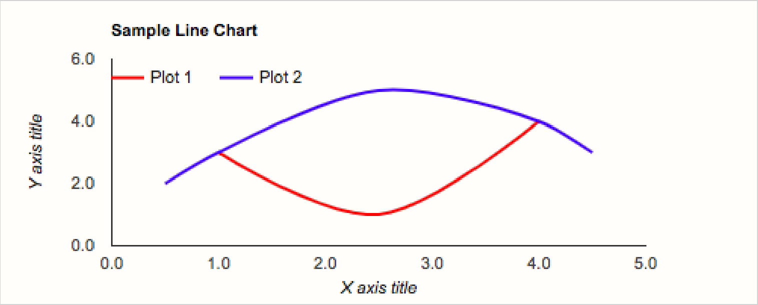
The material charts are still "in beta" - which means they are not finished. Simple charts seem to work well and look great, but some of the options that work for classic charts don't work in material charts, and some charts just don't work properly at all in material mode. It's easy to switch between the two modes, so you can try your charts in both modes to see which works best and which you prefer.

Material charts do not support downloading as PNG images and if you configure a PNG download for a material chart, you'll see an error below the chart.

Presumably the material charts will improve over time.

LINE CHARTS AND AREA CHARTS

Line charts can have up to 20 plots, with each plot representing one line on the chart. Area charts are very similar, but have the area under the lines filled with colour.



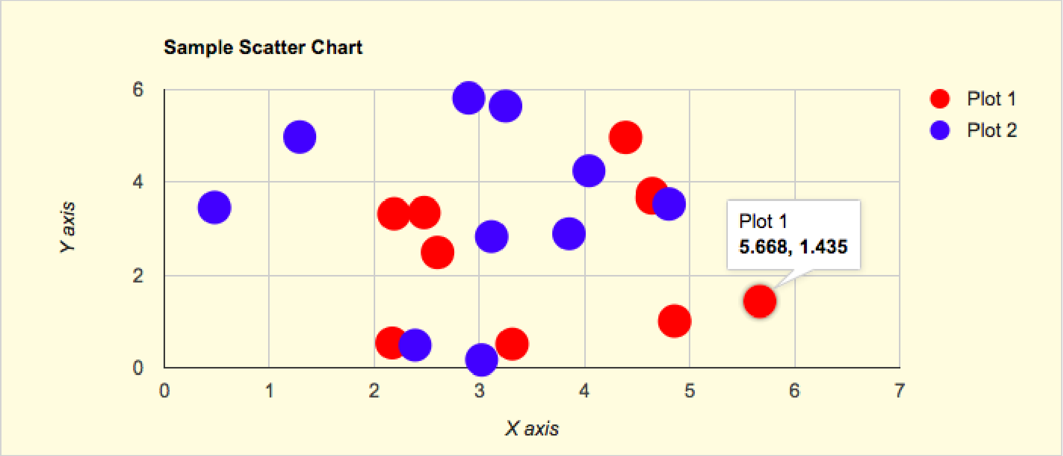
Query Requirements	**Column 1: Numeric - the X value** **Column 2: Numeric - the Y value** The query for each plot must return two columns, the first being the X, or horizontal, axis, and the second being the Y, or vertical axis. Both values must be numeric. If the X axis relates to dates or time, it must be a Unix time value.
Multiple Plots	**Yes** Each plot has its own SQL query, a colour, a name, and a line style.
Axis Overrides	**Both X and Y** You can override the automatic axis limits calculated from the data series. The example above extends the Y axis from 0 to 30°C.
Data Formatting	**Both X and Y** You can apply formatting to the X and Y axis labels. For example if the X axis data represents Unix time (seconds since 1st January 1970), you can use the formatting options to display many different date or time formats. In the example above, the X axis is formatted to show hours. There are also many numeric formats to choose from.
Extra Columns	**Yes** annotation, annotationText, certainty, emphasis, interval, scope, tooltip

Plots and plot options are discussed further in the section on Plot Options. Axis Overrides are discussed further in the section on Axis Options.

For area charts with multiple plots, all the plots must return exactly the same set of X values. The underlying reason is that the Google Charts API does not currently support "interpolateNulls:true" for area charts, as it does for line charts. Failure to adhere to this restriction will result in incorrect charts being drawn.

SCATTER GRAPHS

A scatter graph is a set of individual dots. Scatter charts are used to show groupings or patterns in data points that have two variables.

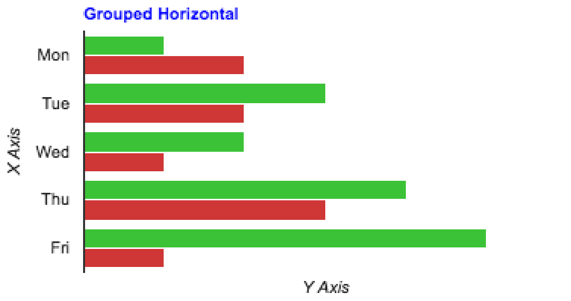


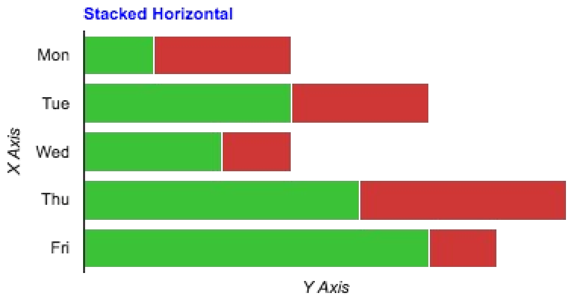
Query Requirements	**Column 1: Numeric - the X value** **Column 2: Numeric - the Y value** The query for each plot must return two columns, the first being the X, or horizontal, axis, and the second being the Y, or vertical axis. Both values must be numeric. If the X axis relates to dates or time, it must be a Unix time value.
Multiple Plots	**Yes** Each plot has its own SQL query, a colour, a name, and a line style.
Axis Overrides	**Both X and Y** You can override the automatic axis limits calculated from the data series and extend the axes to values of your choice.
Data Formatting	**Both X and Y** You can apply formatting to the X and Y axis labels. For example if the X axis data represents Unix time (seconds since 1st January 1970), you can use the formatting options to display many different date or time formats. In the example above, the X axis is formatted to show months. There are also many numeric formats to choose from.
Extra Columns	**Yes** certainty, emphasis, scope, tooltip

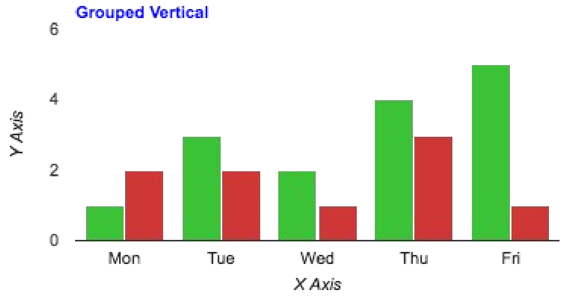
BAR CHARTS

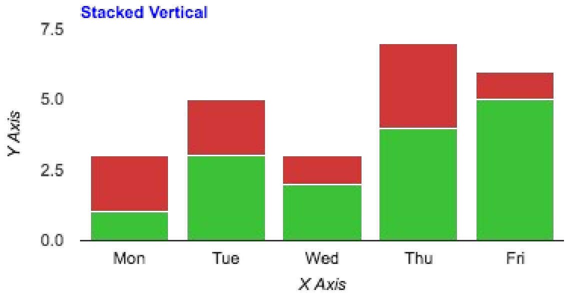
Bar charts are useful when the values of one axis are associated with a named item. Bar charts can be grouped or stacked, horizontal or vertical. The four examples here are all drawn using the same data. Only the chart type has been changed.

Notice that the X and Y axes switch positions for horizontal and vertical bar charts. This means that when you are setting up your chart you can ignore whether it will be drawn horizontally or vertically.









Query Requirements	**Column 1: Text** **Column 2: Numeric** The query for each plot must return two columns, the first being the texts for the X axis, and the second being the values for the Y axis. You need to make sure the text returned in the first column fits along the X axis, using SQL date or string functions to format the text appropriately.
Multiple Plots	**Yes** Each plot has its own SQL query, a colour, a name, and a line style.
Axis Overrides	**Y Axis Only** You can override the Y axis limits calculated from the data series. Note that for stacked bar charts you will usually want to set Y minimum to zero.
Data Formatting	**Y Axis Only** Your query must format the X axis text to suit your requirements. The Y axis can be labelled automatically or with the number and format of labels you choose.
Extra Columns	**Yes** certainty, interval, scope, tooltip

Bar Chart Ordering

Consider these three datasets:

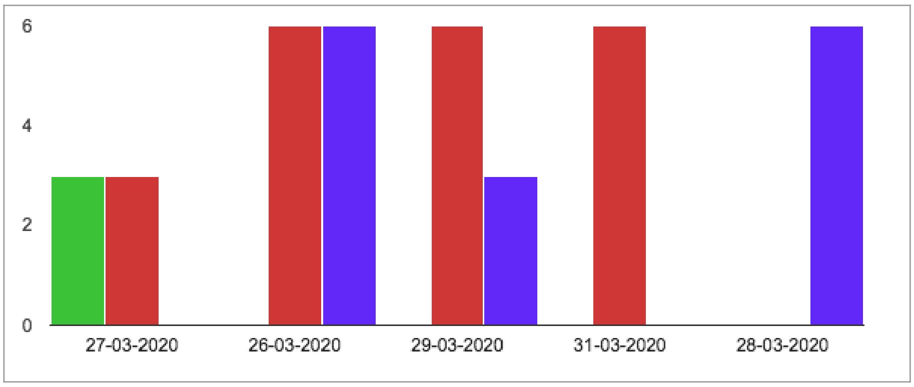
Date	Amount
27-03-2020	3

Date	Amount
26-03-2020	6
27-03-2020	3
29-03-2020	6
31-03-2020	6

Date	Amount
26-03-2020	6
28-03-2020	6
29-03-2020	3

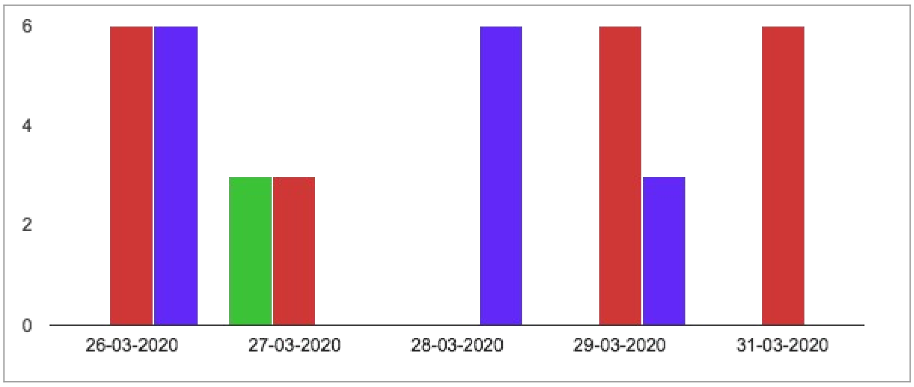
Each dataset is sorted in ascending date order **BUT** not all dates are represented in all datasets, so the combined data is not in ascending date order.

By default, the bars will appear in the order that their values are encountered:



For some kinds of data this would be the expected behaviour.

When the X axis consists of dates, it obviously makes more sense to draw the bars in the correct order, so you can use Plotalot's barchart ***ordered*** option to sort the merged data before the chart is drawn:



Beware, this is ordering by ascending text value, not by date, so you will need to make sure your chosen string date format enables this to work correctly.

Drawing a Bar Chart with Only One Data Value

A table of historical dive information includes the maximum depth of the dive. We want a bar chart to show the relative frequency of dives to various depths.

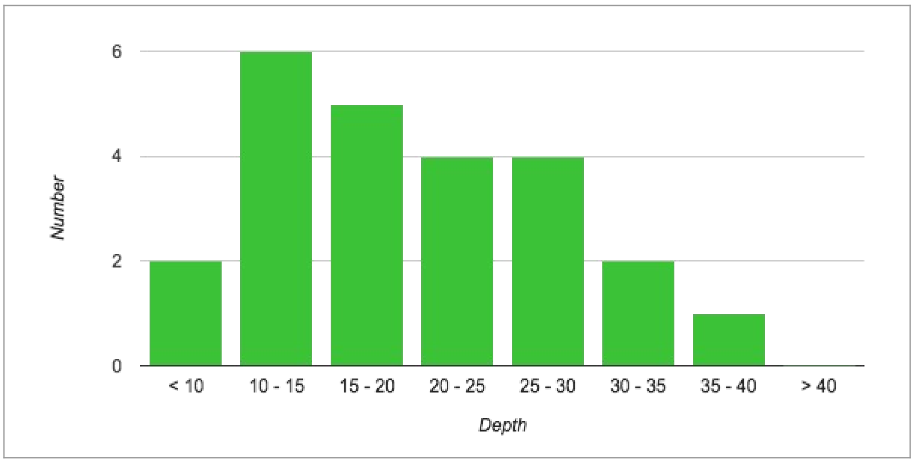
For this purpose, only a single column of the table is relevant, the maximum depth. These are the values of that column from 24 rows:

13, 20, 26, 25, 22, 9, 13, 12, 18, 11, 32, 24, 19, 28, 32, 25, 22, 18, 15, 12, 9, 10, 18, 37

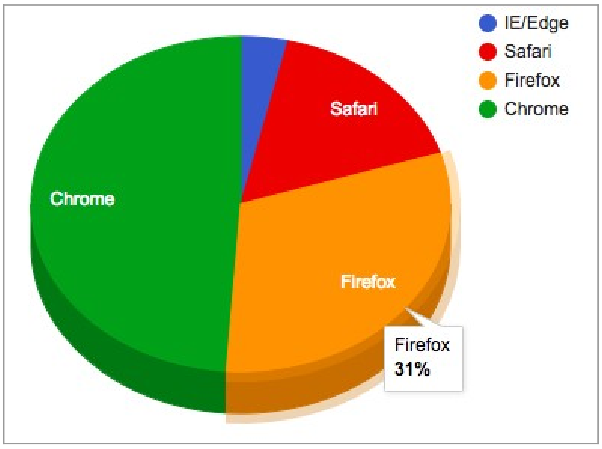
In this example, we generate each bar of the graph with a separate query, and use UNION to return each result as a separate row of a single result set. The descriptive text for each bar (the X axis value) is hard coded as the first value of each sub-query.

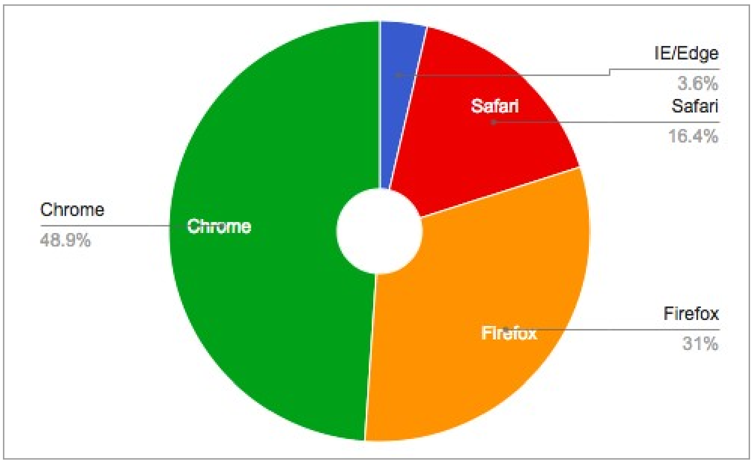
```
SELECT '< 10', SUM(CASE WHEN `max_depth` < 10 THEN 1 ELSE 0 END) AS Y FROM `#__dive_details`
UNION
SELECT '10 – 15', SUM(CASE WHEN `max_depth` >= 10 AND `max_depth` < 15 THEN 1 ELSE 0 END) AS Y
       FROM `#__dive_details`
UNION
SELECT '15 – 20', SUM(CASE WHEN `max_depth` >= 15 AND `max_depth` < 20 THEN 1 ELSE 0 END) AS Y
       FROM `#__dive_details`
UNION
SELECT '20 – 25', SUM(CASE WHEN `max_depth` >= 20 AND `max_depth` < 25 THEN 1 ELSE 0 END) AS Y
       FROM `#__dive_details`
UNION
SELECT '25 – 30', SUM(CASE WHEN `max_depth` >= 25 AND `max_depth` < 30 THEN 1 ELSE 0 END) AS Y
       FROM `#__dive_details`
UNION
SELECT '30 – 35', SUM(CASE WHEN `max_depth` >= 30 AND `max_depth` < 35 THEN 1 ELSE 0 END) AS Y
       FROM `#__dive_details`
UNION
SELECT '35 – 40', SUM(CASE WHEN `max_depth` >= 35 AND `max_depth` < 40 THEN 1 ELSE 0 END) AS Y
       FROM `#__dive_details`
UNION
SELECT '> 40', SUM(CASE WHEN `max_depth` > 40 THEN 1 ELSE 0 END) AS Y FROM `#__dive_details`
```

And here's the chart:



PIE CHARTS AND DONUT CHARTS





Query Requirements	**Column 1: Text** **Column 2: Numeric** The query must return two columns, the first being the slice label text, and the second being the value. The text column is used to name each slice of the pie chart. You can use SQL string formatting functions to manipulate the label text to your requirements.
Multiple Plots	**No** Pie charts can only show a single data series, where each slice represents one data value.
Axis Overrides	**No** The axis limit modifiers are not used for pie charts. The data values are always treated as proportions of the total value of all the rows in the dataset.
Data Formatting	**Yes** You can use Plotalot's Y format selector to format the numeric values.
Extra Columns	**Yes** tooltip

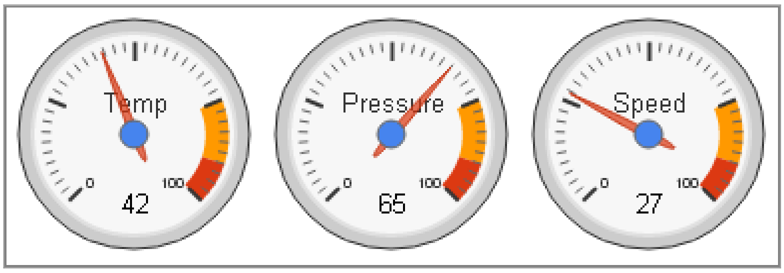
Pie charts can be 2D or 3D, with or without values. The ***with values*** option appends the Y value, in brackets, to the slice label text.

The text shown on the slices can be none, the actual value, the calculated percentage of the whole, or the label.

Donut Charts (pie charts with a hole in the middle) can be created using the pieHole option. The example chart above has the Extra Option ",pieHole: 0.2".

GAUGES

Gauges show a single value. Each row of the dataset draws a single gauge.



Query Requirements	**First Column: Text** **Second Column: Numeric** The query must return two columns, the first being the gauge name, and the second being the value.
Multiple Plots	**No** Gauges only show a single data series.
Axis Overrides	**Y Axis** The Y axis limits are used to set the range of the gauges. If not specified, the default range is 0 - 100.
Data Formatting	**No** Gauge values are always shown as integers.
Extra Columns	**No** The API does not support any extra columns for gauges.

Yellow and Red Zones

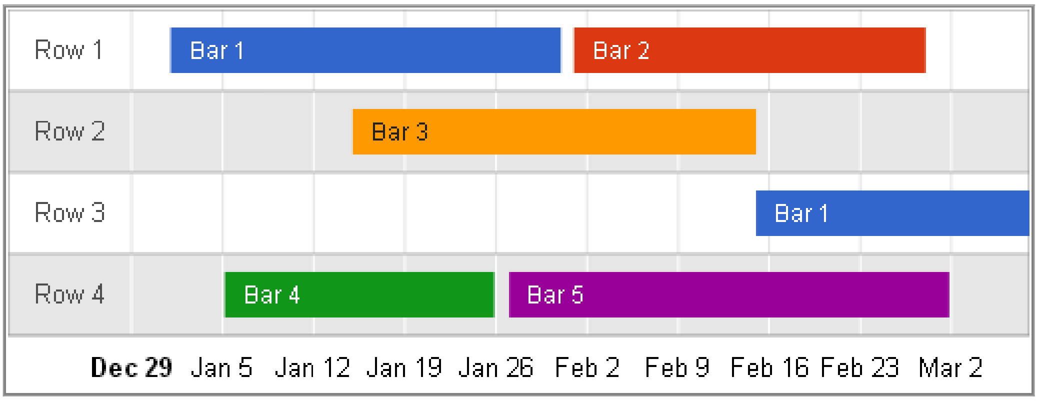
You can set these using the ***extra options*** field. In the example above, the following extra options are specified:

```
,redFrom:90, redTo:100, yellowFrom:75, yellowTo:90, minorTicks:10
```

Refer to the Extra Options section for more details about the syntax.

TIMELINE CHARTS

Timeline charts are good for showing how resources are used over time. They are perfect for overviews of accommodation or meeting rooms, or for resources on a project.

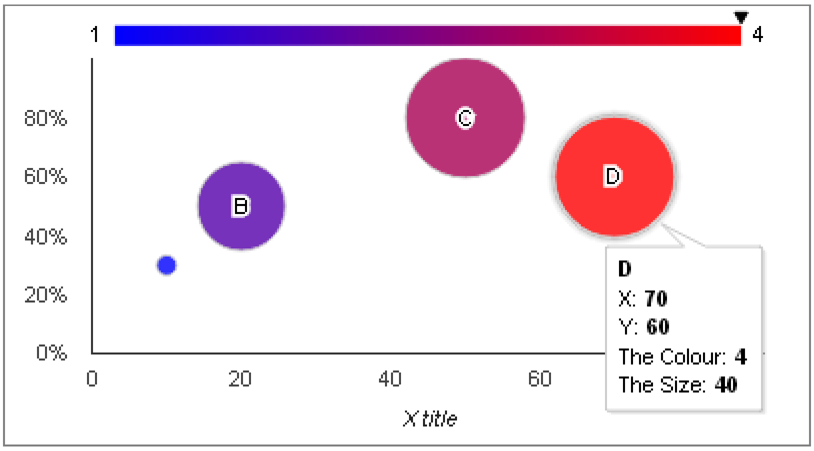


Query Requirements	**Column 1: Text - row name** **Column 2: Text - bar name** **Column 3: From Date** **Column 4: To Date** The query must return all four columns, the first being the row name, the second being the name of the bar. The third column is the start date and the fourth column is the end date. Dates must be retrieved as Unix Timestamps, i.e. using the UNIX_TIMESTAMP( ) sql function.
Multiple Plots	**No** Timelines only show a single data series.
Axis Overrides	**No** The date range is determined by the data.
Data Formatting	**No** Text is shown as returned by the query.
Extra Columns	**Yes** tooltip

Timeline charts don't have a title, and you can't specify data ranges, axis titles, or the date format. You can specify the chart size and background colour.

BUBBLE CHARTS

Bubble charts can represent XY data with size and colour showing two additional dimensions



Query Requirements	**Column 1: Text - the name of the bubble** **Column 2: Numeric - the X coordinate** **Column 3: Numeric - the Y coordinate** **Column 4: Text or Numeric - the series or the colour** **Column 5: Numeric - the size of the bubble** The query must return all five columns. The type of column 4 determines if the chart is a "series" bubble chart or a "gradient-colour" bubble chart.
Multiple Plots	**No** Bubble charts only show a single data series.
Axis Overrides	**Both X and Y** You can override the automatic axis limits calculated from the data series and extend the axes to values of your choice.
Data Formatting	**Both X and Y** You can apply formatting to the X and Y axis labels. For example if the X axis data represents Unix time (seconds since 1st January 1970), you can use the formatting options to display many different date or time formats.
Extra Columns	**No** The API does not support any extra columns for bubble charts.

Column 1 can be an empty string if you don't want the bubbles to have names.

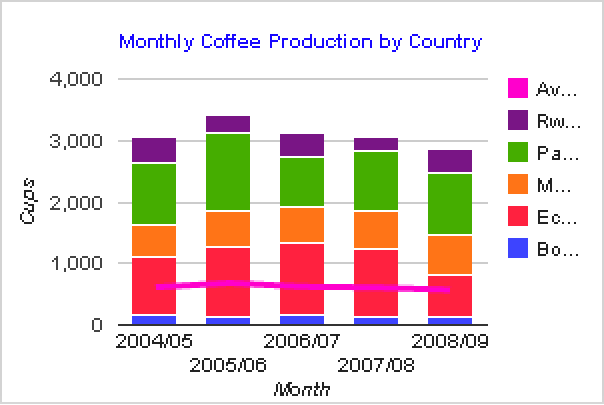
Column 4 can be 0 or an empty string if you want all the bubbles to be the same colour.

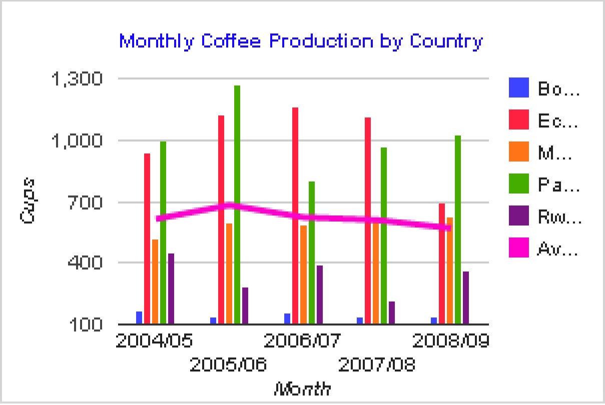
Column 5 can be 0 if you want all the bubbles to be the same size.

The column names of the Query determine the data names shown in the tooltips.

COMBO CHARTS

Combo charts combine several different types of chart together. Each plot can be drawn as bars, lines, area, candlestick, or stepped area. Combo charts can be grouped or stacked. Stacked charts show bar and area data stacked. The examples here are drawn using the same data. The one on the left is stacked, the one on the right is grouped. Only the chart type has been changed.





Combo charts can have text or numeric X axes, but the X values must be consistently text or numeric for all plots. If the X axis data is text, you need to make sure the text fits along the X axis, using SQL date or string functions to format the text appropriately. If the X data is numeric, the X axis formatting options can be used to select the way it is displayed. If the X data represents dates or time, it must be a Unix time value.

The Stepped Area plot type can only be used with text X data. It cannot be use with numeric X data.

Most plot types only require two columns, X and Y. Candlestick plots require 4 columns (minimum, open, close, maximum).

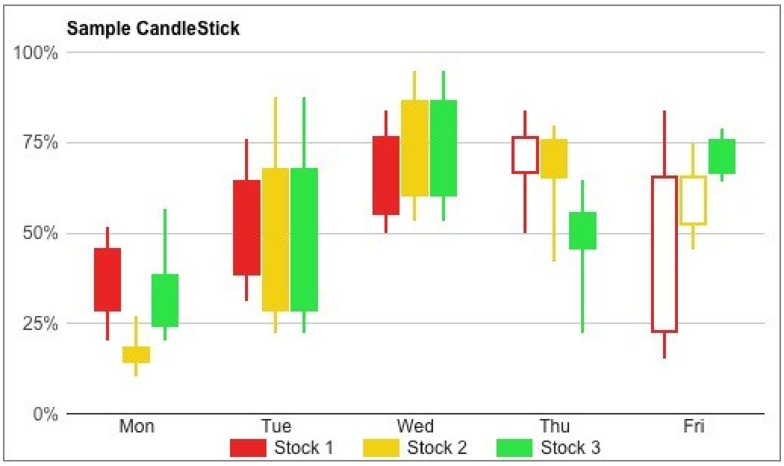
Query Requirements	**Column 1: Text or Numeric - the X axis values** **Column 2: Numeric - the Y values** **Columns 3 - 5: Numeric, for candlestick plots**
Multiple Plots	**Yes**
Axis Overrides	**Numeric X and Y** If the X axis data is text, you cannot override the X axis limits.
Data Formatting	**Numeric X and Y** You can apply formatting to the X and Y axis labels. For example if the X axis data represents Unix time (seconds since 1st January 1970), you can use the formatting options to display many different date or time formats. If the X axis data is text, you cannot apply formatting to the X axis labels.
Extra Columns	**Yes** annotation, annotationText, certainty, emphasis, interval, scope, style, tooltip

Hiding Legend Entries

For Combo charts only, you can hide individual legend entries by naming plots with a trailing tilda (~) character.

CANDLESTICK CHARTS

Candlestick charts are used to show daily stock prices, showing the open, close, minimum and maximum values. In this example, items where the open is less than the close (a gain) are drawn as filled boxes, and items where the close is less than the open (a loss) are drawn as hollow boxes.

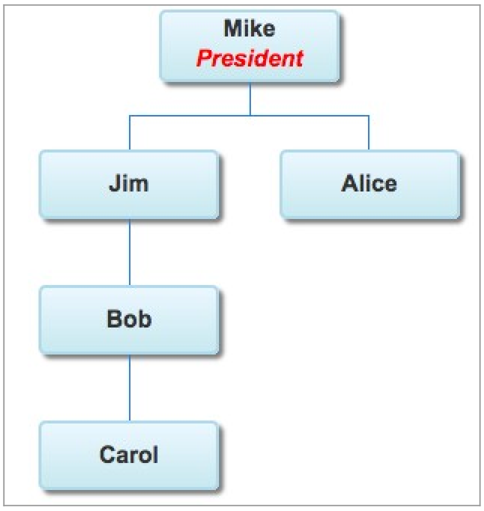


Query Requirements	**Column 1: Text or Numeric - the X axis values** **Column 2: Numeric - the minimum values** **Column 3: Numeric - the opening values** **Column 4: Numeric - the closing values** **Column 5: Numeric - the maximum values** The query for each plot must return at least five columns, the first being the values for the X axis, and the next four being numeric. The X values must be consistently text or numeric for all plots. If the X axis data is text, you need to make sure the text fits along the X axis, using SQL date or string functions to format the text appropriately. If the X data is numeric, the X axis formatting options can be used to select the way it is displayed. If the X data represents dates or time, it must be a Unix time value.
Multiple Plots	**Yes**
Axis Overrides	**Numeric X and Y** If the X axis data is text, you cannot override the X axis limits.
Data Formatting	**Numeric X and Y** You can apply formatting to the X and Y axis labels. For example if the X axis data represents Unix time (seconds since 1st January 1970), you can use the formatting options to display many different date or time formats. If the X axis data is text, you cannot apply formatting to the X axis labels.
Extra Columns	**Yes** tooltip

You can select "ordered" to sort the X axis, although this only affects charts that use text for the X axis. See "Bar Chart Ordering" for more information.

ORGANISATION CHARTS

Organisation charts show hierarchical structure with a single root. They can't, unfortunately, be used for family trees because they can only have one root, and the branches cannot merge.

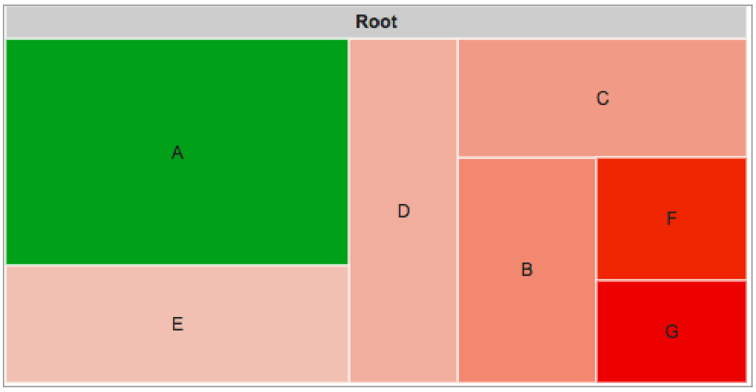


Query Requirements	**Column 1: Text - name** **Column 2: Text - manager** **Column 3: Text - (optional) tooltip** The query must return at least the first two columns, the third is optional. For the topmost node, the manager column should be an empty string.
Multiple Plots	**No**
Axis Overrides	**No**
Data Formatting	**No**
Extra Columns	**No** The tooltip column is optional but no extra column needs to be defined.

The "Cell Object" syntax can be used to provide formatting for the names. In the example above, the first node name is defined as "{v:'Mike', f:'Mike<div style=\"color:red; font-style:italic\">President</div>'}".

If you use HTML in the formatted name or in the tooltip you must specify the extra option ",allowHtml:true".

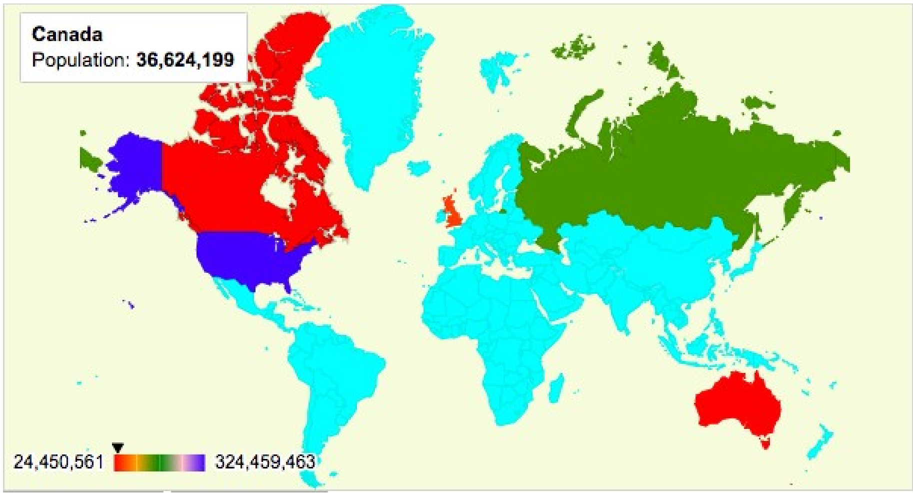
TREE MAP CHARTS



Tree map charts are interactive. Left clicking on a node displays its child nodes. Right clicking moves back up the tree.

Query Requirements	**Column 1: Text - name** **Column 2: Text - parent** **Column 3: Text - size** **Column 4: Text - (optional) colour** The query must return at least the first three columns, the fourth is optional. The root node should have a blank parent column.
Multiple Plots	**No**
Axis Overrides	**No**
Data Formatting	**No**
Extra Columns	**No** The Google Charts API does allow custom tooltips but they are not supported by Plotalot for this chart type.

GEO CHARTS



Plotalot support Geo Charts in four modes:

- Region Mode shows each defined region in a separate colour, with a numeric value that determines the region's colour, and can be used to convey numeric data for the region. The example above is Region Mode.
- Markers by Address Mode places markers at named locations on the map with tooltips that can convey numeric data.
- Markers by Latitude/Longitude Mode is like Markers by Address Mode but uses coordinates instead of names.
- Text Mode shows the location name on the map.

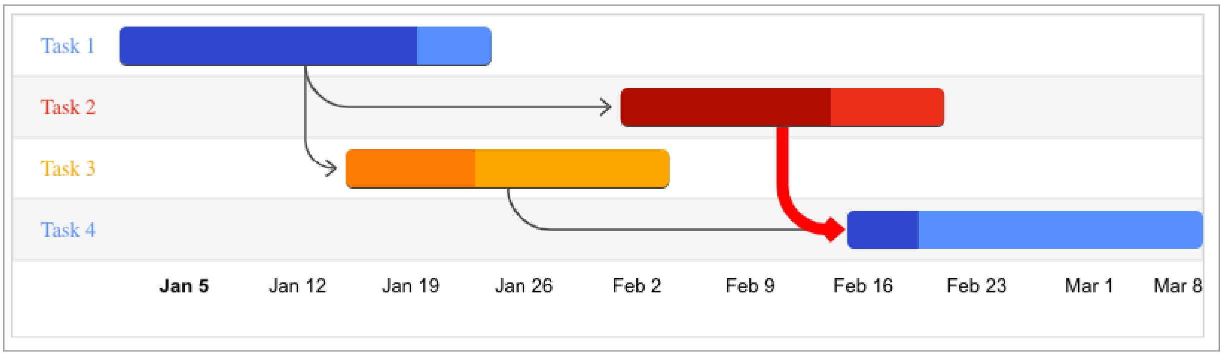
Query Requirements	**Region Mode**	**Column 1:** Text location **Column 2 - optional:** Numeric value for relative colour
	Markers by Address	**Column 1:** Text location **Column 2 - optional:** Numeric value for relative colour **Column 3 - optional:** Numeric value for relative marker size
	Markers by Latitude / Longitude	**Column 1:** Numeric latitude **Column 2:** Numeric longitude **Column 3 - optional:** Numeric value for relative colour **Column 4 - optional:** Numeric value for relative size
	Text Mode	**Column 1:** Text location **Column 2 - optional:** Numeric value for relative text size
Multiple Plots	**No**	
Axis Overrides	**No**	
Data Formatting	**No**	
Extra Columns	**Yes** tooltip	

You need to obtain a free Google Maps API Key for this chart type:

https://developers.google.com/maps/documentation/javascript/get-api-key

GANTT CHARTS

Gantt charts show project tasks and their dependencies, and optionally, the critical path.

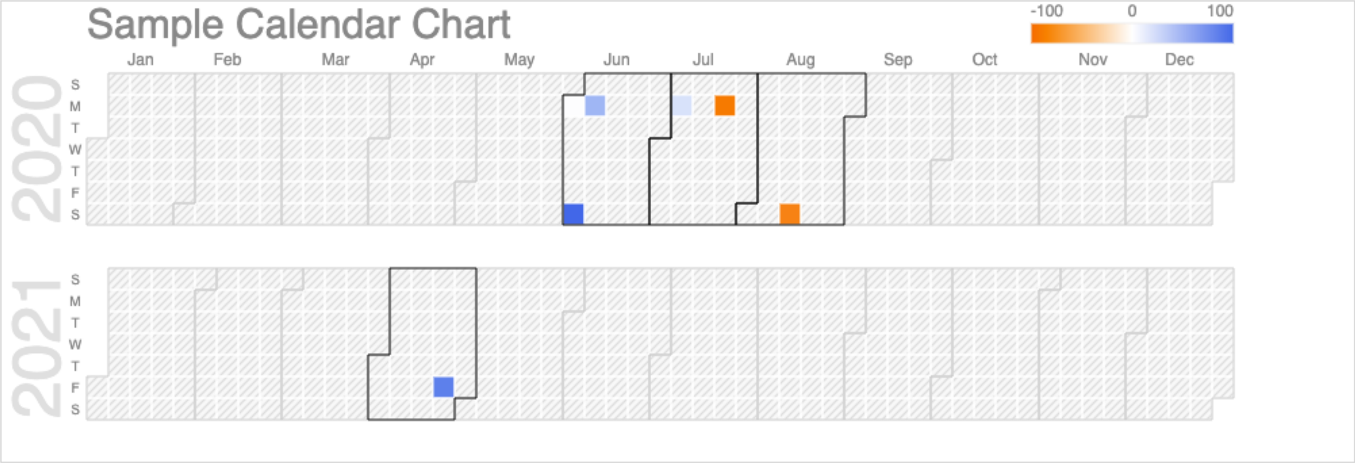


Query Requirements	**Column 1: Text - Task ID** **Column 2: Text - Task Name** **Column 3: Text - Resource ID** (or null or an empty string) **Column 4: Numeric - Start Date as a Unix Timestamp** (or 0) **Column 5: Numeric - End Date as a Unix Timestamp** (or 0) **Column 6: Numeric - Duration in milliseconds** (or 0) **Column 7: Numeric - Percentage complete** **Column 8: Text - Dependencies as a comma separated list of Task ID's** (or null or an empty string) The query must return eight columns. Some columns may be omitted by specifying 0, null, or an empty string. At least two of columns 4, 5 and 6 must be specified.
Multiple Plots	**No**
Axis Overrides	**No**
Data Formatting	**No**
Extra Columns	**No**

The Dependencies column must not include any references to non-existent Task ID's. Any such references will result in a red 'error' message instead of a chart. Tasks that have no dependencies should specify the Dependencies column as null or an empty string. Space (' ') is also accepted as a null dependency.

CALENDAR CHARTS

Calendar charts show daily activity or volume over months or years.

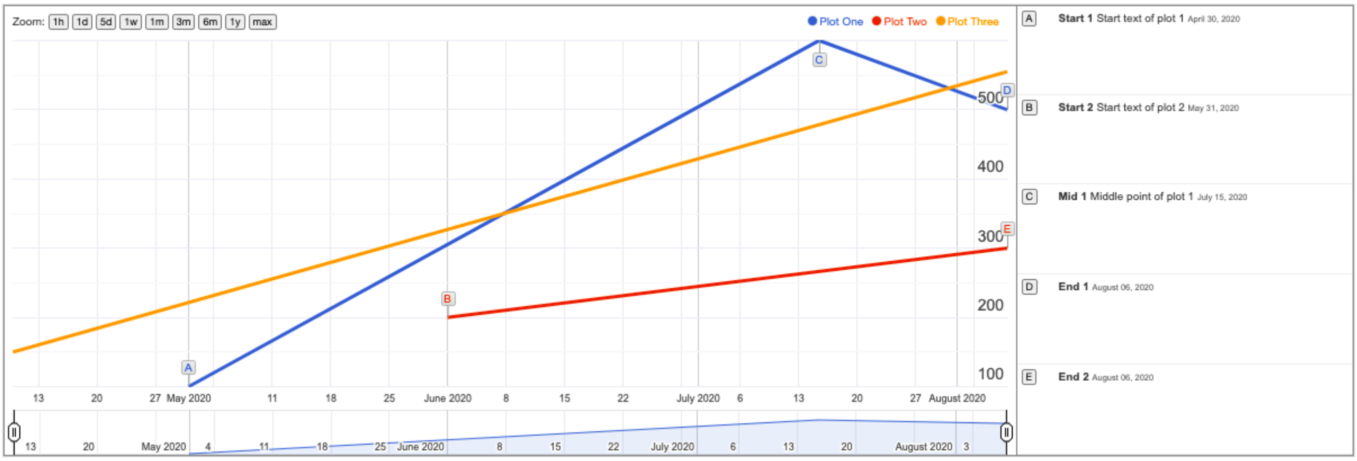


Query Requirements	**Column 1: Numeric - Date as a Unix Timestamp** **Column 2: Numeric - A value that determines relative colour.** An optional third column for customized styling is coming in a future Google Charts release.
Multiple Plots	**No**
Axis Overrides	**No**
Data Formatting	**No**
Extra Columns	**No**

Calendar charts are not responsive. For use on small screens you need to enclose it in a scrollable container element.

ANNOTATION CHARTS

Annotation charts are timeline charts with annotations shown in full on the right hand side.



Query Requirements	**Column 1: Numeric - Date as a Unix Timestamp** **Column 2: Numeric - The Y value** **Column 3: Text - Annotation Title** (or null or an empty string) **Column 4: Text - Annotation Detail** (or null or an empty string) The query must return four columns, the first being a number representing a Unix timestamp.
Multiple Plots	**Yes** Each plot has its own SQL query, a colour, and a name.
Axis Overrides	**Y Axis** You can override the automatically calculated axis limits and extend the Y axis to values of your choice.
Data Formatting	**No**
Extra Columns	**No**

AXIS OPTIONS

X and Y Axis Titles

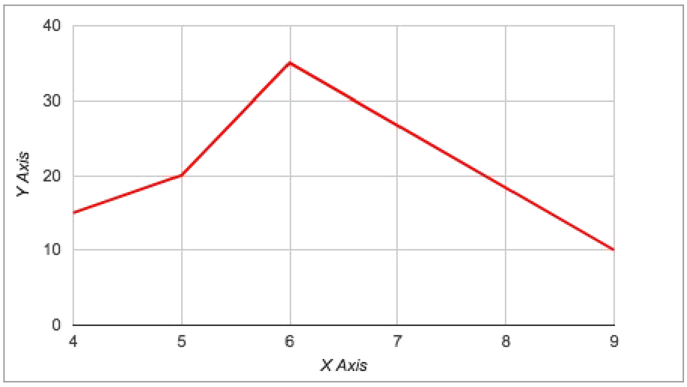
The X and Y axis titles are optional labels that you can add to the chart. They can be fixed text, or the results of SQL queries.

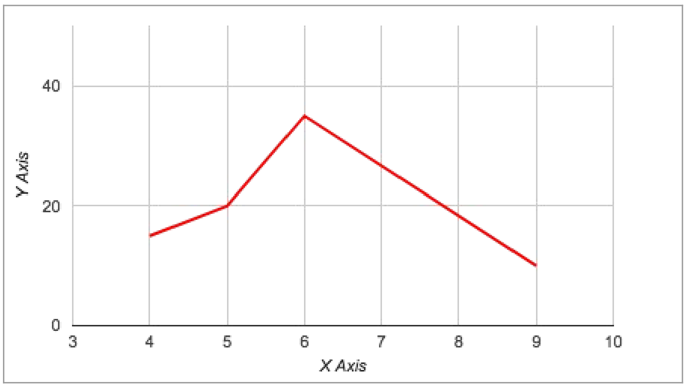
Axis Start and Axis End

If you leave the axis start and axis end options blank, the chart axes are determined automatically. The axis start and axis end options can be used to extend the chart's visible range. Here's a small dataset to illustrate.

X	4	5	6	9
Y	15	20	35	10

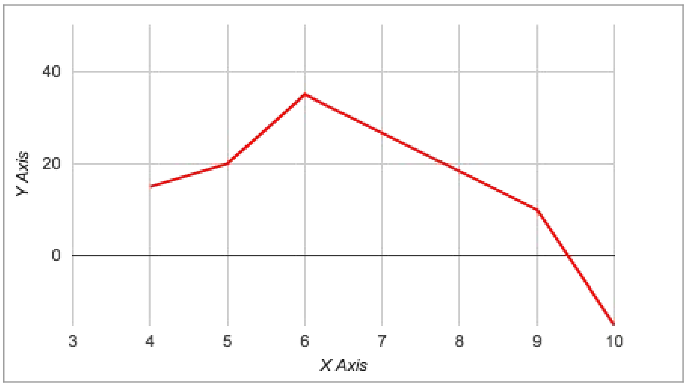
The chart on the left does not specify axis limits, so they have been set automatically. The chart on the right sets the X axis limits at 3 - 10 and the Y axis limits at 0 - 50:





You can specify the limits as fixed numbers, or the result of SQL queries.

If the plot data extends outside the specified limits, Plotalot ignores the specified values and use the limits of the actual data instead. In such cases a warning is shown in the chart editor (warnings are never shown on the front end of the site). You can use this feature to set "preferred" axis values, which dynamically extend when necessary. Let's add a new row to the dataset above, X = 10, Y = -15, but keep the axis limits set the same as in the right hand chart above.



Now Plotalot generates a warning in the chart editor:

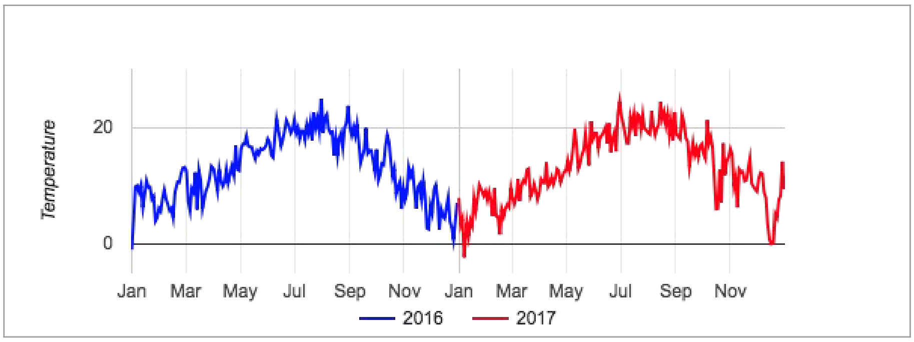
Y start [0] > first Y value in data [-15]

And extends the Y start value to -15.

More About Data Ranges

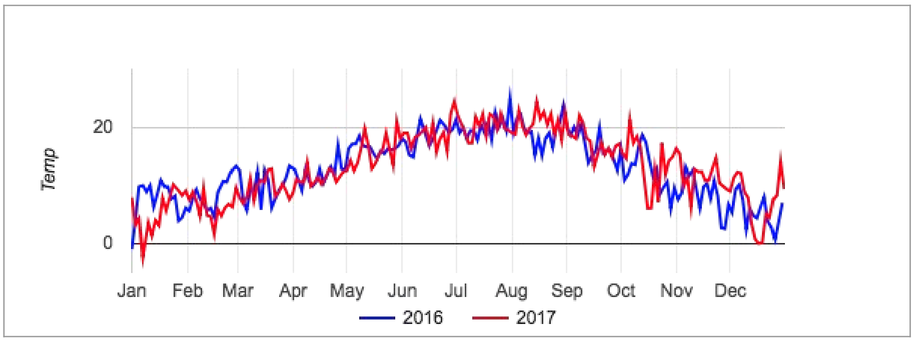
Plotalot analyses the data returned by the query (or queries) to determine the axis ranges for the chart. You can also extend the range of the chart using the axis range options.

For charts with multiple plots, the axis ranges of the chart are the entire range of all the data returned by all the plots. For example, this chart shows temperature readings for two different years. The X axis range covers the whole two years:



In this case, we can "trick" Plotalot into plotting both years on top of each other by arranging the queries so that they both return dates in the same date range. For example, you could add a year to the dates returned by one of the plots:

```
SELECT UNIX_TIMESTAMP(CAST(DATE_ADD(date, INTERVAL 1 YEAR) AS DATETIME)),…
```



If you use this trick, you will probably need to switch off the tooltips because the values shown in them may not make proper sense. You can do this with the "enableInteractivity" extra option.

X Labels and Y Labels

These options determine the number of labels and gridlines on each axis. The default is -1, which means that the number of labels will be decided automatically, which is useful if the data is very variable and the number of labels needs to be dynamic. To set a fixed number of labels, specify an integer greater than zero.

Label Formats

These options determine the format of the axis labels.

Numeric Label Formats

The numeric formats ("99,999.9" etc) format numbers using the thousands and decimal separators of the user's browser locale.

There is an exception to this for pie charts with values. For pie charts only, the numeric values are formatted by Plotalot on the server, which does not know the user's browser locale. So for the Y axis formats, you will notice that there are more numeric formats ("99,999.9", "99 999,9", etc). For most charts there will be no difference between "99,999.9" and "99 999,9". But for the numeric values of pie charts, there will.

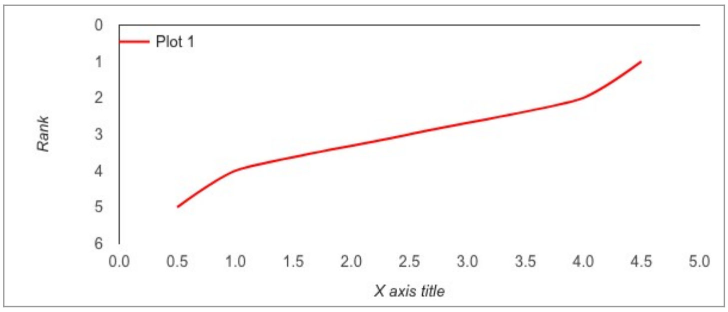
Inverted Axes

Most chart types have a "vAxis.direction" option:

https://developers.google.com/chart/interactive/docs/gallery/linechart

You can use this in the chart's Extra Options (see below) to make charts with inverted axes. For example:

```
,vAxis:{title:'Rank',gridlines:{color:'transparent',count:7},format:"#,##0",direction:-1}
```



Date Label Formats

The date and time options require the source data to be a Unix timestamp. For example, a query for a chart of temperatures for a range of dates might be:

```
SELECT UNIX_TIMESTAMP(Date), Value FROM Temperature_History
```

You can then use any of Plotalot's date or time formats to format the X axis labels.

The date and time format options only work correctly if the data value being formatted is a true Unix DateTime value, that is, the number of seconds since the 1st of January 1970. They won't work correctly for any other type of date or time value. You need to make sure your SQL query returns date/times as **Unix** DateTimes.

Custom Date Format

One of the X label formatting options is "Custom Date". When you select it, a field appears for you to enter a custom date format, according to the rules given here:

https://unicode-org.github.io/icu/userguide/format_parse/datetime/#date-field-symbol-table

For example: "yy-MM-dd hh:mm". The custom format only works for dates and times. You can't use it to format ordinary numbers.

PLOT OPTIONS

Some types of chart can only show one data series, for example pie charts. But line graphs and bar charts can show several sets of data on the same graph. For these charts, Plotalot lets you specify multiple plots.

To change the number of plots, just change the number of plots in the Chart Options and click Save.

Each plot has four or five properties.

Name	The name is used as the legend label for the plot. It can be text or a SQL query.

Colour	The colour used for the plot. If you leave this blank the colour is decided automatically.
Style	Options for the lines on line graphs and the colouring of pie charts.
Enable	Enable toggles the plot between enabled and disabled modes. Disabled plots are not drawn.
Query	The large text area is where you write the SQL query that returns data for the plot.

EXTRA OPTIONS

The Google Charts API has far too many options for them all to be accommodated in a reasonable user interface, and they are continually being added to. So Plotalot just shows a manageable number of options that can satisfy many common needs, but it does not stop you from using the full power of the Google API. The ***Extra Options*** field can be used to add additional Google Charts options to a chart, or override the options chosen by Plotalot. All of the options are described in the Google Charts API documentation.

To find the relevant options, go to the Google Chart Gallery, then click on the chart type you are working with, and scroll down to "Configuration Options". Some options are common to several charts, but it's always best to look at the options for the specific chart you are dealing with.

Plotalot will quite happily work with many options that we have never tried, or even heard of. However, please note that we can't guarantee it will work with every option.

Please don't ask us questions about Extra Options. We have some experience of some of the options, but we don't know about all of them. The correct place to ask detailed questions about API Options is:

https://groups.google.com/forum/#!forum/google-visualization-api

Option Syntax

The format of the ***Extra Options*** field is critical. Any errors will cause a Javascript error, and the chart will not be drawn.

Plotalot's main job is to construct the Javascript that will draw your chart. You can see the Javascript constructed in the Script tab. Part of the script is the options. Here's an example from one of the sample charts. We have re-formatted it to make it easier to read.

```
var options = {
title:'Sample Stacked Bar Chart',
width:1200,
height:400,
backgroundColor:'#E6E5E5',
titleTextStyle:{color:'#0000FF'},
isStacked:true,
series:[{color:'00CC00'},{color:'FFFF00'},{color:'FAA500'}],
legend:{position:'bottom'},
hAxis:{title:'',gridlines:{count:16},format:'#,##0'},
vAxis:{title:'',gridlines:{color:'transparent'}}
<extra options are added here>
};
```

Plotalot chooses the options and parameters to add based on your selections in the chart editor. Each option is separated by a comma. Any extra options are added at the end of the options chosen by Plotalot. When you add extra options, you must always place a comma **before** each new option, to separate it from the previous option. You must not place a comma after the last option because it will cause an error in some older browsers.

Simple Options

Let's examine the options shown above in detail. Some options are easy, for example:

```
title:'Sample Stacked Bar Chart'
```

If you look up the "title" option in the Google documentation, you will see that it is described as a "string". String values must be enclosed in single or double quotes.

Width and height are numbers. They do not need quotes. isStacked is a boolean, which is either true or false, again without quotes.

Objects

Now look at titleTextStyle. Notice the different construction, with the curly brackets:

```
titleTextStyle:{color:'#0000FF'}
```

If you look up "titleTextStyle" in the documentation, you will see that it is described as an "Object", and that it has more options than just the colour. An Object is a container for several options. Objects can be nested within other Objects (you may have noticed that the whole "options" list is in fact an Object). The curly brackets are Javascript notation for an Object. In fact, titleTextStyle can contain several more options:

```
{color: <string>, fontName: <string>, fontSize: <number>}
```

So we could increase the title font size of the chart by adding this string in ***Extra Options***:

```
,titleTextStyle:{fontSize: 18}
```

If you added that extra option and examined the chart script again, you would now see two entries for titleTextStyle :

```
titleTextStyle:{color:'#0000FF'},
…
titleTextStyle:{fontSize: 18}
```

Extra Options are always added at the end of all the options generated by Plotalot, so they always override the options chosen by Plotalot. In this case, the second titleTextStyle Object completely replaces the first, which means that the color option from the first titleTextStyle Object is no longer active. Although the text is now larger, it is no longer blue. If we want to keep both options, we would have to re-specify the color in the second titleTextStyle Object, like this:

```
,titleTextStyle:{color:'#0000FF', fontSize: 18}
```

Some options are Objects with more Objects nested inside them. Notice that the hAxis option is an Object, and that the gridlines option inside it is also an Object.

Just remember that if you want to override any property of an object, you must re-specify the entire object. It's easy enough once you get used to it, but you must take care to get the curly brackets exactly right.

Arrays

There is one more construct you may need to know about. Look at the series option above, and notice the square brackets:

```
,series:[{color:'00CC00'},{color:'FFFF00'},{color:'FAA500'}]
```

If you look up "series" in the documentation, you will see that it is described as an "Array of objects, or object with nested objects". The square brackets are Javascript notation for an Array. In fact, the Objects have several options that you could use, for example:

```
,series:[{color:'00CC00',curveType:'function'},
   {color:'FFFF00',visibleInLegend:false,lineWidth:'3'},
   {color:'FAA500'}]
```

Documentation Syntax

Javascript's curly bracket syntax is not very documentation-friendly, so the Google documentation uses the "dot" notation to describe the properties and structure of Objects, for example:

```
chartArea.left
```

This means that "left" is a member of the "chartArea" Object. You need to write this in ***Extra Options*** using Javascript Object notation, like this:

```
,chartArea:{left:'10%'}
```

Examples

We'll now give a few examples of useful ***Extra Options***. There are many more possibilities.

titleTextStyle

This option applies to all chart types that have a title, and controls the style of the main chart title. You can use html colour names or # colour notation.

```
,titleTextStyle:{color:'blue', fontName:'Comic Sans MS', fontSize:18}
```

hAxis.titleTextStyle and vAxis.titleTextStyle

These options control the style of the axis titles. Note that since hAxis and vAxis are also used by Plotalot, you must repeat the options that Plotalot sets for these options.

```
,hAxis:{title:'X Axis Title',
   gridlines:{count:16},format:'#,##0',textStyle:{color:'white'},
   titleTextStyle:{color:'lightgray', fontName: 'Tahoma', fontSize: 11}}
```

hAxis.textStyle and vAxis.textStyle

These options control the style of the axis labels. Note the comments above about Plotalot's use of these options.

```
,vAxis:{textStyle:{color:'lightgray'}}
```

backgroundColor.stroke and backgroundColor.strokeWidth

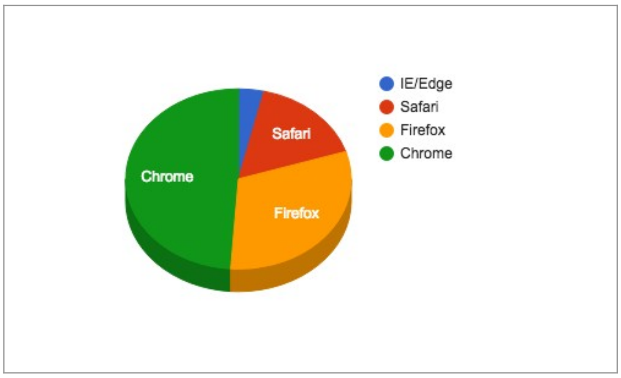
Controls the chart border.

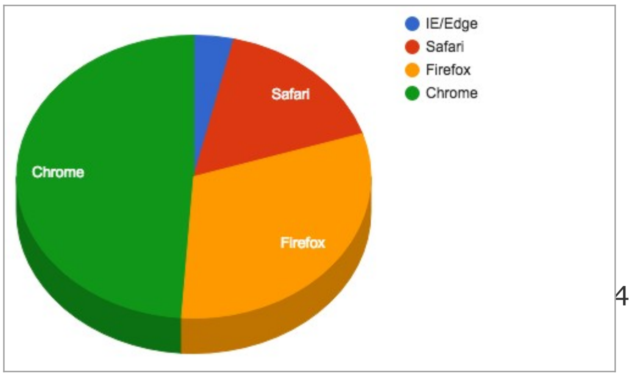
```
,backgroundColor:{stroke:'gray',strokeWidth:2}
```

chartArea

This option applies to most chart types and is very useful for controlling the spacing around the chart. It makes a huge difference to pie charts. These two charts have the same dimensions, but the one on the right has:

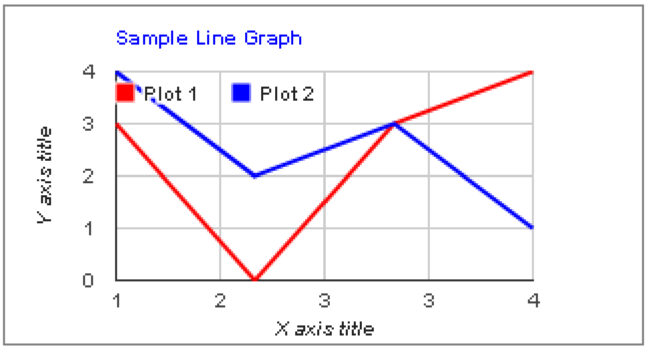
```
,chartArea:{top:'3%',width:'100%',height:'100%'}
```

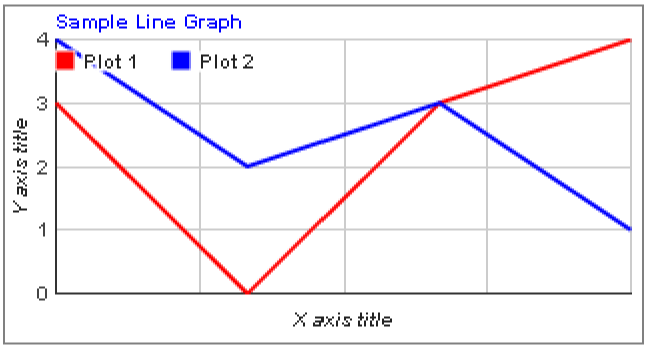




With some chart types, giving the chart area 100% of the drawing area leaves no room for titles or axis labels, so you need to adjust the option values to get the best results in each case.
These two charts have the same dimensions, but the one on the right has:

```
,chartArea:{left:'8%',top:'10%',width:'90%',height:'75%'}
```

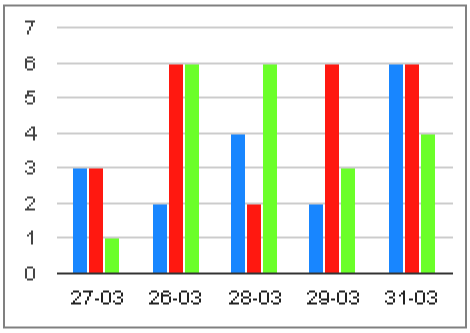


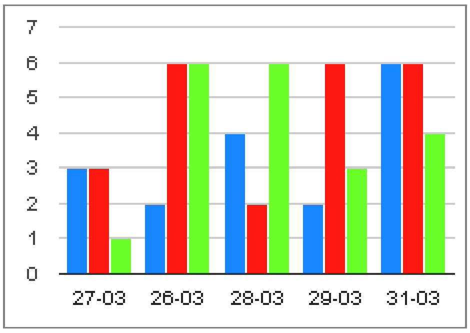


bar.groupWidth

This option applies only to bar charts, and controls the width of a group of bars, with the width specified in pixels, or a percentage of the available width for each group, where '100%' means that groups have no space between them. The chart on the left does not have this option. The chart on the right does:

```
,bar:{groupWidth:'85%'}
```



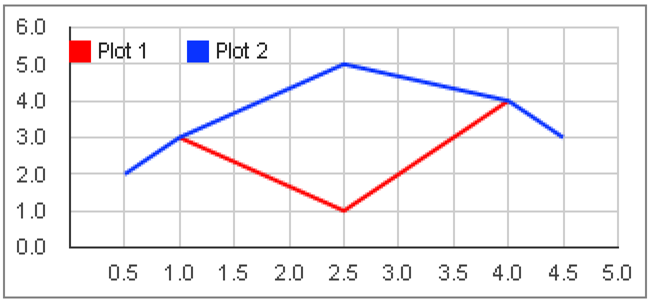


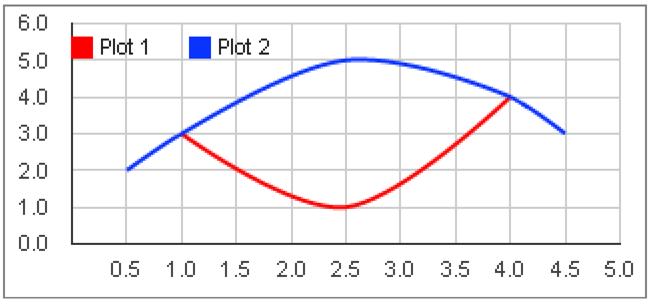
curveType

This option applies only to line charts, and controls the curve of the lines.

```
,curveType:'function'
```

It works well for line charts that don't have many data points, creating smooth curves. The chart on the left does not have this option. The chart on the right does:





enableInteractivity

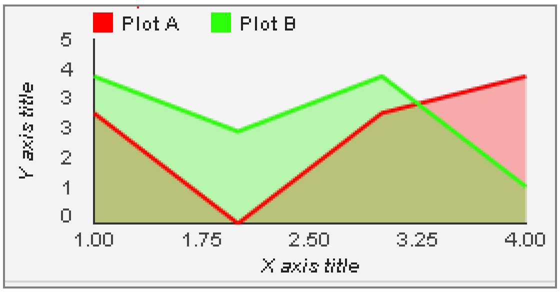
You can switch off the tooltips for a chart using this option.

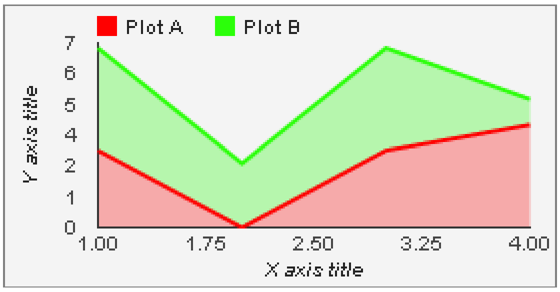
```
,enableInteractivity:false
```

isStacked

You can create a stacked area chart using this option. The chart on the left does not have this option. The chart on the right does:

```
,isStacked:true
```





Trendlines

Many chart types support automatically drawing trendlines, for example:

```
,trendlines: { 0: {}}
```

Check the documentation at https://developers.google.com/chart/interactive/docs/gallery/trendlines

Dynamic Extra Options

If the Extra Options field begins with "Select", it is assumed to be a SQL query, and will be resolved to form the extra options string.

EXTRA COLUMNS

For some chart types, the Google Charts API supports extra columns that tell the API more about how to draw a chart. The Google documentation calls this feature "DataTable Roles":

https://google-developers.appspot.com/chart/interactive/docs/roles

Plotalot allows you to specify which extra column types your query will return, and will then pass the data to the API in the correct format.

The extra columns supported by Plotalot are:

Name	Data type
annotation	string
annotationText	string
certainty	number (0=false, 1 = true)
emphasis	number (0=false, 1 = true)
interval	number

scope	number (0=false, 1 = true)
tooltip	string
style	string

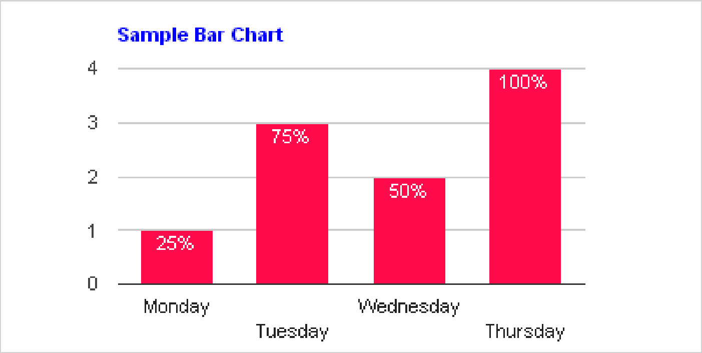
Example 1 - Bar Chart Annotations

Let's add some annotations to the Sample Bar Chart. First we specify one extra column, "annotation". This tells Plotalot that our query will supply one additional string column after the two normally returned for a bar chart. Plotalot will now require that your query returns three columns.

Then we'll add the extra column to the sql. For this example the values are hard coded, but of course you can use any combination of SQL functions to calculate and format the values. Note that since the annotation data type is "string", the query must return the column value as a string.

```
SELECT 'Monday', 1,'25%'
      UNION SELECT 'Tuesday', 3,'75%'
      UNION SELECT 'Wednesday', 2 ,'50%'
      UNION SELECT 'Thursday', 4, '100%'
```

Here's the result:



Example 2 - Annotations on a Line Chart

Now we'll add two extra columns to the Sample Line Chart. We will specify "annotation, annotationText" (without the quotes) in the Extra Columns field. This tells Plotalot that our query will supply two extra columns after the two normally returned for a line chart. Plotalot will now require that your query returns four columns.

The first extra column is the annotation that will appear on the chart, and the second is the contents of the tooltip that shows when you hover the mouse over the annotation. Here's the new sample data:

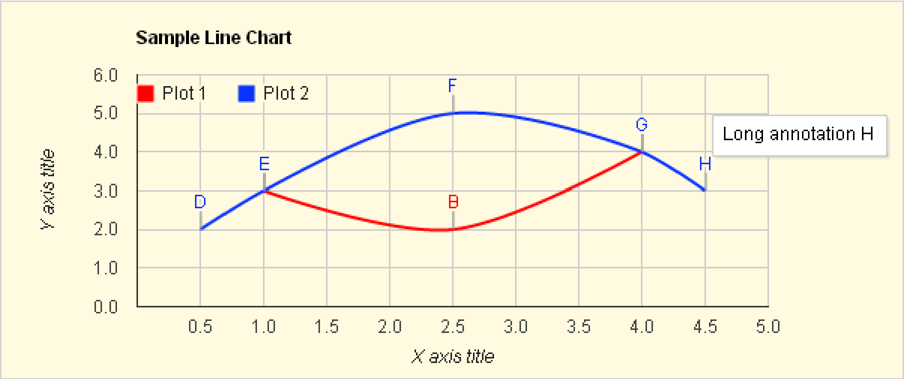
Plot 1:

```
      SELECT 1, 3, 'A', "Long annotation A"
            UNION SELECT 2.5, 2, 'B' , "Long annotation B"
            UNION SELECT 4, 4, 'C', "Long annotation C"
```

Plot 2:

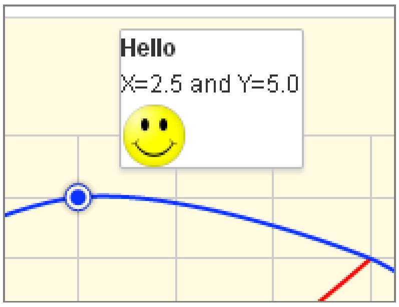
```
      SELECT 0.5, 2, 'D', "Long annotation D"
            UNION SELECT 1, 3, 'E', "Long annotation E"
            UNION SELECT 2.5, 5, 'F', "Long annotation F"
            UNION SELECT 4,4, 'G', "Long annotation G"
            UNION SELECT 4.5,3, 'H', "Long annotation H"
```

Here's the result, hovering the mouse over annotation H.



Example 3 - HTML Tooltips

Let's add some tooltips to the sample line chart.



- We need to add the extra column "tooltip".
- We also need the extra option ",tooltip: {isHtml: true}".

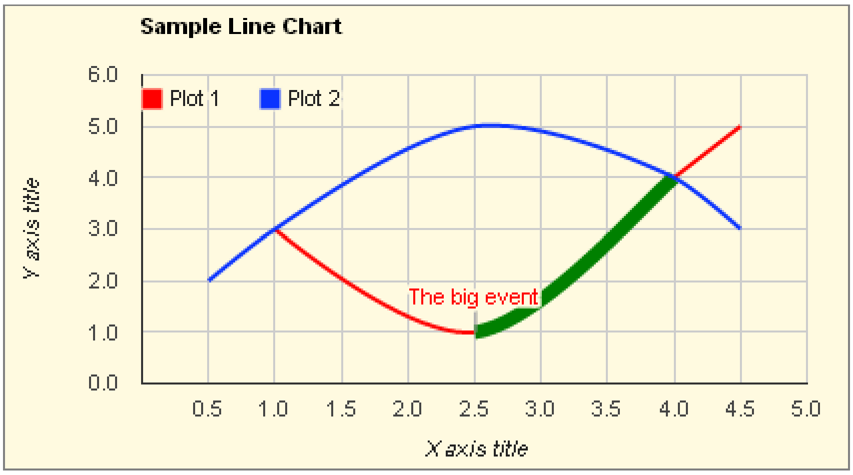
Here's the data for one of the rows:

```
UNION SELECT 2.5, 5,'<strong>Hello</strong><br />X=2.5 and Y=5.0<br />
      <img src="/images/smile.png" alt="">'
```

Example 4 - Style

The style role can change the styling of bars, lines and data points. In this example we added the style "stroke-width:7; stroke-color:green" to one of the data points.

We also added an annotation "The big event" to the previous data point.



ERRORS AND WARNINGS

The display area of the chart editor shows any errors or warnings that apply to your chart.

Plotalot Warnings

Warnings tell you about a setting that you have made that has been ignored because it cannot be applied to the current chart. Warnings do not prevent the chart being drawn. They are shown for information. They are never shown when the chart is drawn on your website. The most common warnings are when axis overrides cannot be applied because they are inconsistent with the retrieved data. For example:

```
Warning: Y start [0] > first Y value in data [-6.25]
```

This warning is saying that you have asked for the chart's Y axis to start at the value zero, but Plotalot is ignoring your request because the lowest value in the relevant dataset is -6.25. You can read more about this in the section on Axis Options.

Plotalot Errors

Errors prevent the chart being drawn. There are two main types of errors, Plotalot errors and SQL errors. This is a typical Plotalot error:

```
Error: For a Line graph, the first two columns must be numeric.
Plot 2 column 1 [Date] is non-numeric, and column 2 [Value] is numeric
```

This error has occurred because the SQL query for plot 2 has returned the wrong column type(s) for the selected chart type. In this case, a line graph was selected, which requires two numeric columns, but the query returned a non-numeric column called Date and a numeric column called Value. The chart cannot be created until you correct the query to return the correct types of columns.

This is a typical SQL error:

```
Error: Plot 2: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'DATE(CURDATE())) ORDER BY UnixDateTime' at line 1 [1064]
```

The SQL error code [1064] is included at the end of the error message. Sometimes it helps to Google the error number, for example, "SQL 1064".

The chart cannot be created until you correct the error.

Javascript Errors

Javascript errors are detected by your browser, and indicate a problem parsing the chart script. The most likely cause of a Javascript error is an error in your ***Extra Options***.

A typical Javascript error looks like this:

```
Javascript error: Uncaught SyntaxError: Unexpected token }
```

Clear the ***Extra Options*** field, and if the error goes away, check your option syntax carefully.

Tables

Plotalot can draw tables in three different ways.

Plotalot Tables	HTML generated by Plotalot using <table>, <tr>, and <td> elements. No Javascript. Not responsive. Cannot be dynamically sorted by column. No CSV download. Plotalot can add a totals row. Easy to re-style. Suitable for small tables that do not need to be responsive.
Responsive Tables	HTML generated by Plotalot using <div> elements with responsive CSS. No Javascript. Highly responsive. Cannot be dynamically sorted by column. No CSV download. Plotalot can add a totals row.
Google Tables	Generated in the browser using Javascript provided by Google. Responsive using scrolling. Clicking column headings sorts by column. Can add a CSV download button. Extra Options include paging, frozen columns, etc. Cells can have their own scroll bars. Plotalot cannot add a totals row.

COMMON FIELDS

The fields of the table editor change depending on the table type. Some fields are common to all types.

Name

The table name is just for your reference. It is not displayed with the table generated.

Type

Plotalot, Responsive, or Google.

Row Numbers

Whether to include a row number column as the first column of the table.

Database

To use a database other than the current Joomla site you must first create it using the "Databases" menu. You will then be able to select it here.

Max Rows

You may want to limit the number of rows shown on your web page. If you want the first 10 rows you can use the SQL LIMIT function that we introduced in the SQL section. But what if you need to show a representative sample of a large dataset? You can't easily do that in SQL, but that's exactly what Max Rows does. You specify how many rows you want, and Plotalot samples the data set to display the number of rows you asked for. Leave Max Rows blank to show all rows.

Totals

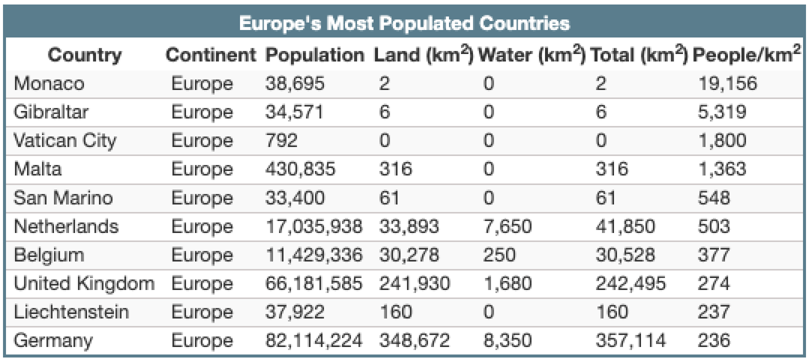
See Table Totals below.

Query

The query can be any valid select query. The table will include the relevant number of rows and columns.

PLOTALOT TABLES

These are HTML tables using <table>, <tr>, and <td> elements. The HTML is constructed by the Plotalot code running on the server, and is delivered to the user's browser in the usual way. No Javascript is involved.



Europe's Most Populated Countries

Country	Continent	Population	Land (km^2)	Water (km^2)	Total (km^2)	People/km^2
Monaco	Europe	38,695	2	0	2	19,156
Gibraltar	Europe	34,571	6	0	6	5,319
Vatican City	Europe	792	0	0	0	1,800
Malta	Europe	430,835	316	0	316	1,363
San Marino	Europe	33,400	61	0	61	548
Netherlands	Europe	17,035,938	33,893	7,650	41,850	503
Belgium	Europe	11,429,336	30,278	250	30,528	377
United Kingdom	Europe	66,181,585	241,930	1,680	242,495	274
Liechtenstein	Europe	37,922	160	0	160	237
Germany	Europe	82,114,224	348,672	8,350	357,114	236

The following additional fields are shown for Plotalot tables.

Title

This can be fixed text or a SQL query. If the result is non-blank, it is added to the table as the top row.

Column Headings

This option will either include the column names as headings or not, as you choose. Your query can modify the heading names using the AS clause.

```
SELECT `product_name` AS `Product`, FORMAT(`net_price`, 2) AS `Net Price`
```

Classes and Styling

The table editor lets you enter your own class names for the various elements of Plotalot tables. The sample table uses the class names shown, which have default styles defined in the supplied plotalot.css file.

Name	Html Element to Which the Class is Applied	Sample Table Class Name
Table Style	<table>	plots1
Title row style	<th>	title
Heading row style	<tr>	heading
Odd row style	<tr>	odd
Even row style	<tr>	even

If you apply the sample table class names to your tables, your tables will be styled like the sample table above.

If your front end site template loads the Bootstrap CSS, you can create well formatted tables simply by setting the table class to "table table-bordered table-striped table-condensed width-auto".

The Plotalot front end CSS file is loaded for Plotalot tables, so you can add your own styling there.

Responsive Behaviour

Plotalot tables are not responsive so are best used for small tables with only a few columns.

RESPONSIVE TABLES

These are also HTML tables, but instead of <table>, <tr>, and <td> elements, they use <div> elements with classes that correspond to the table, rows and cells. The supplied front end CSS file provides the styling. The HTML is constructed by the Plotalot code running on your server, and is delivered to the user's browser in the usual way. No Javascript is involved. In their full width format, responsive tables look exactly the same as Plotalot tables.

The following additional fields are shown for Responsive tables.

Title

This can be fixed text or a SQL query. If the result is non-blank, it is added to the table as the top row.

Column Headings

This option will either include the column names as headings or not, as you choose. You can modify the heading names as described for Plotalot tables.

Width and Height

If you leave these fields blank, the table takes as much space as it needs. If you enter numbers (in pixels) in either of these fields, the table is contained within the dimensions specified, with scrollbars if necessary. In responsive "narrow" mode, the width and height are ignored.

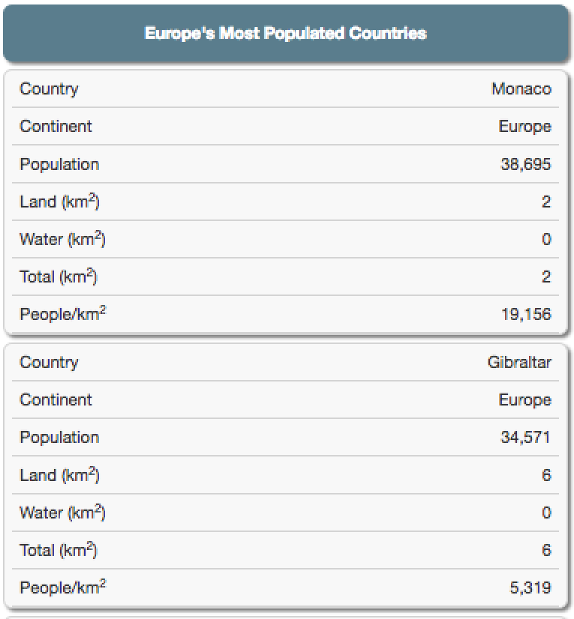
Classes and Styling

The table editor lets you enter a class name for the outermost <div> element of the table, which enables you to style tables individually. The names of the classes used for the inner table, rows and cells are fixed. You can re-style them as required by changing them in the Plotalot CSS file, or override them based on the outer div class. Use your browser developer tools to discover the classes applied to the various elements.

The Plotalot front end CSS file is loaded for Responsive tables, so you can add your own styling there.

Responsive Behaviour

On larger screens, the supplied styling makes Responsive Tables look exactly the same as the Plotalot sample table. But on small screens, Responsive Tables completely change their layout, with each row becoming a separate list, in a format that works well on even the smallest of screens



Europe's Most Populated Countries

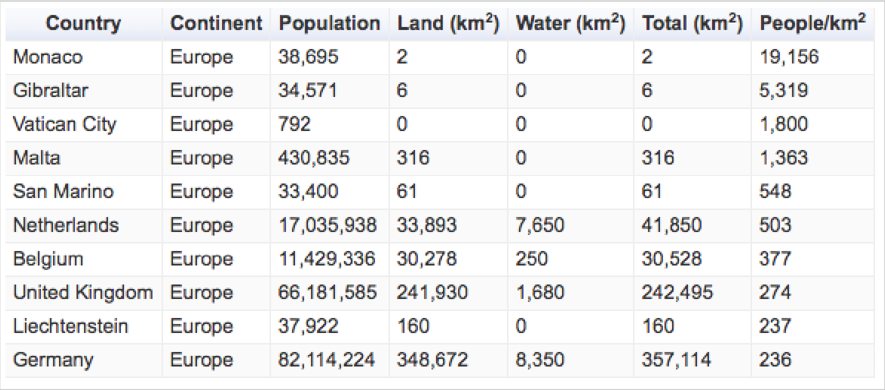
Country	Monaco
Continent	Europe
Population	38,695
Land (km^2)	2
Water (km^2)	0
Total (km^2)	2
People/km^2	19,156

Country	Gibraltar
Continent	Europe
Population	34,571
Land (km^2)	6
Water (km^2)	0
Total (km^2)	6
People/km^2	5,319

GOOGLE TABLES

Google tables are constructed using the Google Charts API. Plotalot constructs the Javascript and the Javascript data object, and the Google Javascript running in the user's browser constructs the HTML using <table>, <tr>, and <td> elements. These tables provide scrolling, paging, sorting, and other active features.

Clicking on a column heading dynamically sorts the rows by that column. Numeric columns sort numerically if the data in the column is numeric. Note that numbers that include commas are treated as strings and will not sort as you would expect. See the Number Formatting section below for details.



Country	Continent	Population	Land (km^2)	Water (km^2)	Total (km^2)	People/km^2
Monaco	Europe	38,695	2	0	2	19,156
Gibraltar	Europe	34,571	6	0	6	5,319
Vatican City	Europe	792	0	0	0	1,800
Malta	Europe	430,835	316	0	316	1,363
San Marino	Europe	33,400	61	0	61	548
Netherlands	Europe	17,035,938	33,893	7,650	41,850	503
Belgium	Europe	11,429,336	30,278	250	30,528	377
United Kingdom	Europe	66,181,585	241,930	1,680	242,495	274
Liechtenstein	Europe	37,922	160	0	160	237
Germany	Europe	82,114,224	348,672	8,350	357,114	236

The following additional fields are shown for Google tables.

Width and Height

If you leave these fields blank, the table takes as much space as it needs. If you enter numbers (in pixels) in either of these fields, the table is contained within the dimensions specified, with scrollbars if necessary.

Extra Options

If you can find your way around the Google Charts API Reference documentation, you can manually add additional options here. There are more details and examples in the Extra Options section. Plotalot cannot validate the options you enter, so you can easily create Javascript errors here.

You could enable page-forward and page-back buttons with:

```
,page:'enable'
```

CSV Link

If you have the Plotalot Plugin, you can add a link below the table to download a Comma Separated Variable .csv file containing the table data. The link can include HTML, so it can be something like <button>Download CSV</button>. Leave this field blank for no CSV link.

Classes and Styling

Google tables assign default class names to most table elements. The table editor lets you enter your own class names to replace the default Google classes, in case you need different styles in different tables.

Note that the <table> element itself always has class "google-visualization-table-table". It can't be changed.

Name	Element	Default Class
Heading row	<tr> elements in <thead>	google-visualization-table-tr-head

Odd rows	<tr> elements of odd rows in <tbody>	google-visualization-table-tr-odd
Table row	<tr> elements of even rows in <tbody>	google-visualization-table-tr-even
Selected row	<tr> of the selected row	google-visualization-table-tr-sel
Hover row	<tr> of the row hovered over	google-visualization-table-tr-over
Header cell	<td> elements in <thead>	google-visualization-table-th
Table cell	<td> elements in <tbody>	google-visualization-table-td
Row number cell	The first <td> of each row	google-visualization-table-seq

If you only have one table, or if you want all of your Google tables to have the same style, there is no need to override the default class names. You can just make your own CSS styles override the Google styles.

The Plotalot front end CSS file is loaded for Google tables, so you can add your own styling there.

Google tables can be tricky to re-style. Use your browser developer tools to see the classes applied to the various elements, and the CSS precedence required to override them.

Examples

Override the background colour for the odd and even rows:

```
tr.google-visualization-table-tr-even {background-color: pink}
tr.google-visualization-table-tr-odd {background-color: yellow}
```

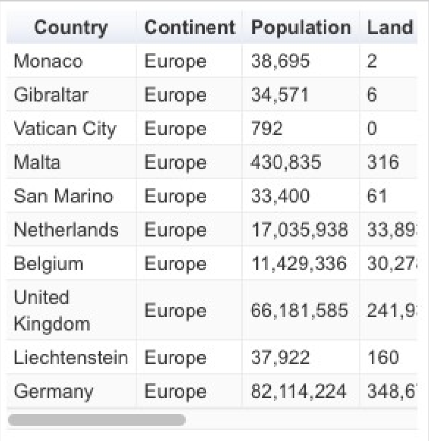
Override the background colour for the heading:

This is more complex because Google assigns the class "google-visualization-table-tr-head" to the <tr> element, and "google-visualization-table-th gradient" to the <th> elements. To assign new colours and remove the gradient, we need:

```
tr.google-visualization-table-tr-head {background-color:black;color:white}
tr.google-visualization-table-tr-head th.gradient {background-image:none}
```

Responsive Behaviour

On narrower screens, Google tables acquire a horizontal scrollbar, which some people regard as responsive behaviour, and some don't.



Country	Continent	Population	Land
Monaco	Europe	38,695	2
Gibraltar	Europe	34,571	6
Vatican City	Europe	792	0
Malta	Europe	430,835	316
San Marino	Europe	33,400	61
Netherlands	Europe	17,035,938	33,89
Belgium	Europe	11,429,336	30,27
United Kingdom	Europe	66,181,585	241,9
Liechtenstein	Europe	37,922	160
Germany	Europe	82,114,224	348,6

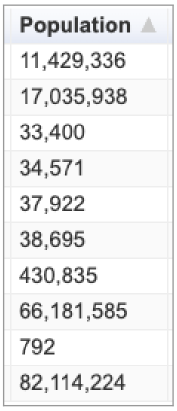
NUMERIC FORMATTING

The easiest way to format numeric data is to use the MySql FORMAT function in your SQL query. For example, to format a number in a typical European format with two decimal places, you could use something like:

```
SELECT FORMAT(`column_name`, 2, 'de_DE');
```

Note that the inclusion of dots and/or commas in the formatted number results in the column being determined as a string column. For Plotalot and Responsive tables this is fine and makes no difference. However for Google tables it does make a difference.

Google tables allow the front end user to dynamically sort columns by clicking on the column headings. When numbers with thousands separators are sorted as strings, the sort sequence is incorrect, as shown on the right.



Population ▲
11,429,336
17,035,938
33,400
34,571
37,922
38,695
430,835
66,181,585
792
82,114,224

For Google tables, you should have your SQL return numeric values unformatted, so that the raw data sent to the browser is numeric. You can then add a custom formatter to the Extra Javascript field, so that the table Javascript can format the numbers correctly for display. Your numeric columns will then sort correctly

To format a number in a typical European format with two decimal places, you would use something like:

```
var formatter = new
google.visualization.NumberFormat({decimalSymbol:',',groupingSymbol:'.'});
formatter.format(window.plotalot_chart_117_data, 3);
```

In this example the chart number is 117 and we are formatting the 4th column (counting from 0, so column 3).

Here's another example, formatting a column as 12 digits without thousands separators:

```
var formatter = new
google.visualization.NumberFormat({pattern:'############'});
      formatter.format(window.plotalot_chart_126_data, 3);
```

The parameters for the NumberFormat function are here:

https://developers.google.com/chart/interactive/docs/reference#numberformat

CONDITIONAL FORMATTING

Sometimes you might want to format values differently depending on their value. You could use SQL to include your formatting in the retrieved data. For example:

```
SELECT `id`,`title`, `language`,
      CONCAT('<span style="color:',
        CASE
        WHEN `hits` BETWEEN 0 AND 100 THEN 'red'
        WHEN `hits` BETWEEN 100 AND 200 THEN 'orange'
        ELSE 'green'
      END,'">',`hits`,'</span>') AS `hits`,
      SUBSTRING(`introtext`,1,40) AS `introtext`, `created`, `modified`
      FROM `#__content`
```

But now the resulting data becomes a string rather than a number, and for Google tables this would mean that the column no longer sorts as expected. A better solution is to use Extra Javascript to add the required formatting to the data, while keeping the column data numeric. In this example the Javascript is more complex because we need to iterate through the Google data table, assigning different formatting to each row:

```
var rows = window.plotalot_chart_103_data.getNumberOfRows();
for (var i=0; i < rows; i++)
   {
   var hits = window.plotalot_chart_103_data.getValue(i, 3);
   if (hits < 100)
      window.plotalot_chart_103_data.setProperties(i,3,
           {style:'background-color:red;color:white;font-style:bold;'});
   if (hits >= 100)
      window.plotalot_chart_103_data.setProperties(i,3,
           {style:'background-color:green;color:white;font-style:bold;'});
   }
```

HYPERLINKS IN TABLES

The sample table shows an example of how to build hyperlinks using the SQL CONCAT function. Here's the SQL used by the sample table. The %%J_ROOT_URI%% variable is used to provide the current site address.

```
SELECT `title` AS `Article`, DATE(`modified`) AS `Updated`,
      CONCAT('<a target="_blank" href="%%J_ROOT_URI%%/index.php?
      option=com_content&view=article&id=', `id`, '">', `title`, '</a>') AS `Link`
      FROM `#__content`
      ORDER BY `modified` DESC
      LIMIT 10
```

TABLE TOTALS

Plotalot can add a totals row to Plotalot tables and Responsive tables. It can't do this for Google tables because the sort function could move the totals row into the body of the table.

Specify a list of columns to total, e.g. 1,2,3 to add totals for the first three columns. For numeric columns, the total is the sum of the row values for that column. For non-numeric columns, the total is the number of rows.

You can optionally specify the number of decimal places, decimal separator, and thousands separator. If you don't specify them they default to 0, ".", and none (prior to version 6.19 they defaulted to 0, ".", and ",").

Use semi-colons (;) to specify these, for example:

`1,2,3`	formats the totals with no decimals and no thousands separators
`1,2,3;2;.;,`	formats the totals with 2 decimal places, decimal separator ".", and thousands separator ","
`1,2,3;0;,;.`	formats the totals with no decimal places and thousands separator "."

You can use SQL number formatting for column values that are to be totalled, provided that the decimal and thousands separators used by the SQL formatting match the separators specified for the totals. Plotalot will use the separators specified to interpret the SQL-formatted numbers. For example, if the decimal and thousands separators specified for the totals are "," and ".", Plotalot will interpret a column value "123.456,789" as the decimal number 123456.789.

Single Items

Single items return a single value from a SQL query with no added formatting, so that when they are used in an article they appear to be part of your article text. For example: “Our download count today stands at 436,991", or "today is Monday the 15th of March". A single value just returns the first column of the first row returned by the query, with no formatting or html at all. Here’s another example:

```
SELECT CASE WHEN HOUR(NOW()) >= 18 THEN "Good evening"
      WHEN HOUR(NOW()) >= 12 THEN "Good afternoon"
      ELSE "Good morning" END
```

There are no rules about the number of columns or their type.

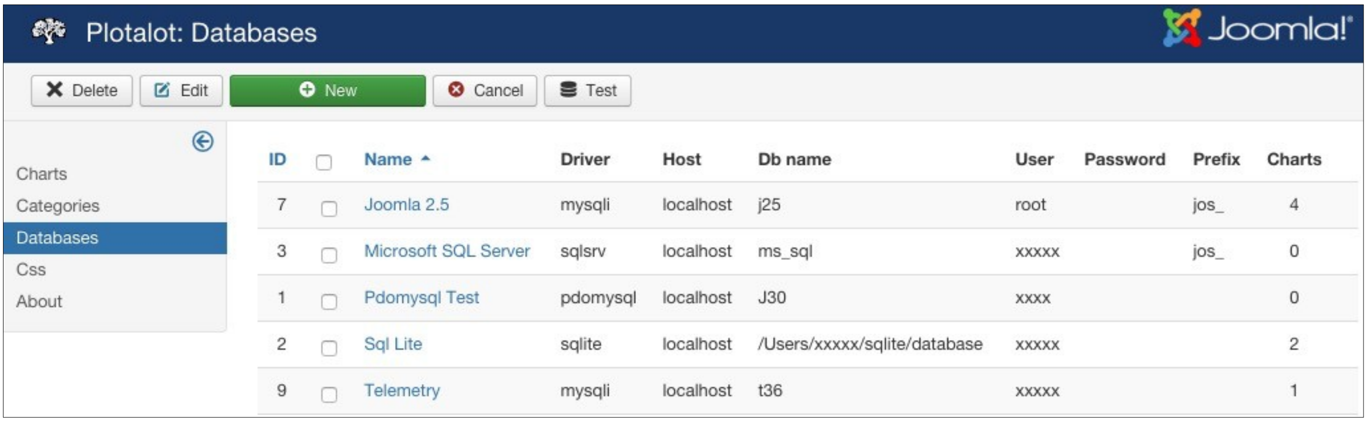
If you set the Plotalot component configuration option to allow non-select queries, you can use a single item to insert or update a database row every time a page is visited. For example:

```
UPDATE `#__hit_counts` WHERE `page` = '51' SET `counter` = `counter` + 1
```

External Databases

The default database for new charts is "This Site", which means that the chart's SQL queries will run against the Joomla site database.

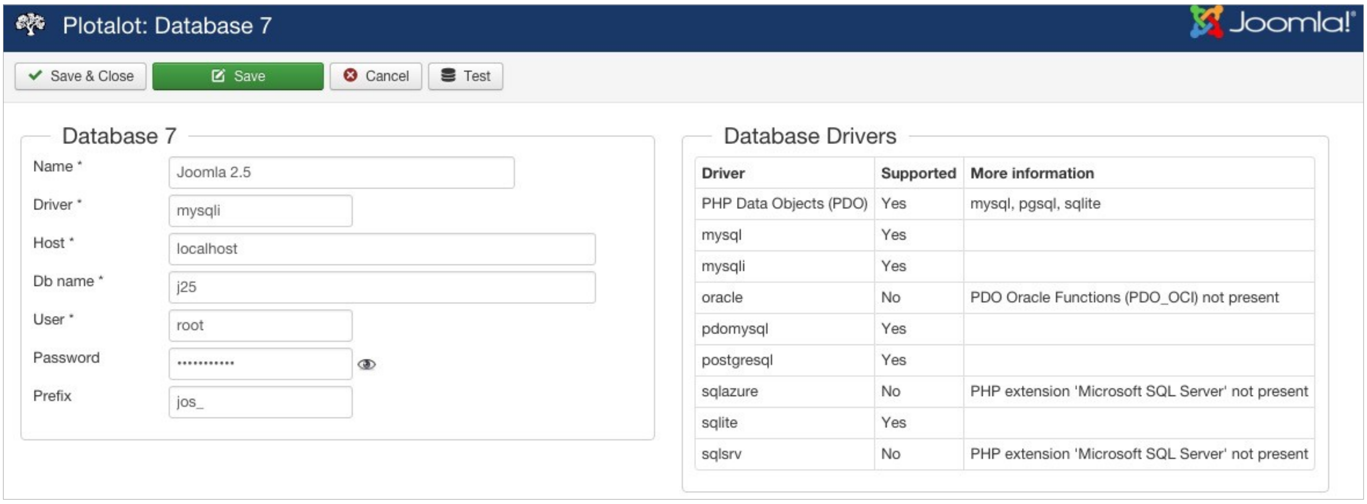
Plotalot can also query external databases. To do that, you first create a record in the "Databases" list, that tells Plotalot how to access the external database.



ID	□	Name ▲	Driver	Host	Db name	User	Password	Prefix	Charts
7	□	Joomla 2.5	mysqli	localhost	j25	root		jos_	4
3	□	Microsoft SQL Server	sqlsrv	localhost	ms_sql	xxxxx		jos_	0
1	□	Pdomysql Test	pdomysql	localhost	J30	xxxx			0
2	□	Sql Lite	sqlite	localhost	/Users/xxxxx/sqlite/database	xxxxx			2
9	□	Telemetry	mysqli	localhost	t36	xxxxx			1

DEFINING A DATABASE

When you create a new database record, you give it a name that you will use to select it in chart definitions. You also supply all of the connection and authorisation parameters. The editor shows a list of the database drivers found on your system, and whether or not your server has the runtime libraries they need.



Database 7

Name *	Joomla 2.5
Driver *	mysqli
Host *	localhost
Db name *	j25
User *	root
Password	•••••••••••
Prefix	jos_

Database Drivers

Driver	Supported	More information
PHP Data Objects (PDO)	Yes	mysql, pgsql, sqlite
mysql	Yes	
mysqli	Yes	
oracle	No	PDO Oracle Functions (PDO_OCI) not present
pdomysql	Yes	
postgresql	Yes	
sqlazure	No	PHP extension 'Microsoft SQL Server' not present
sqlite	Yes	
sqlsrv	No	PHP extension 'Microsoft SQL Server' not present

Name

The name is used to select the database in chart definitions. You can rename database records at any time. They are linked to charts by their internal ID number, not by name.

Driver

Specify the name of a database driver from the right hand panel, for example "oracle".

Host

If a database runs on the same computer as the web server, the host is normally "localhost". If not, you may need to enter its IP address, or its name in the form "db.myhost.com". If the database uses a non-standard port, you can add the port number to the host string, like "db.myhost.com:3307".

Db name

The name of the database on the database server. For Sqlite databases, the database name must be the full absolute pathname of the SqLite database file.

User and Password

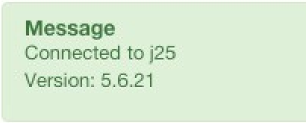
The username and password that will be used to connect to the database. These will be stored in the xxx_plotalot_databases table of the Joomla database. The password is obfuscated (not human readable) but not encrypted, so consider creating a special user with the minimum necessary access rights.

Prefix

The table name prefix for the external database, if there is one. If queries to the database use the "#__" convention, Plotalot will substitute the "#__" for the prefix specified here.

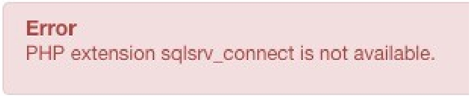
THE TEST BUTTON

When you have saved the database record, you can use the Test button to check that Plotalot can access it correctly. Plotalot will attempt to connect to the database and query the database version. If the query works correctly you should see a green message with the version number.



Message
Connected to j25
Version: 5.6.21

If the connection fails you should see a pink message giving details of the error. In this example, the system does not have the Microsoft Sql Server extension for PHP.



Error
PHP extension sqlsrv_connect is not available.

Component Configuration

You can access the Component Configuration for Plotalot from Joomla Global Configuration or from the Options button on the Plotalot chart list. The Options button is only shown for Super Users.

PERMISSIONS

"Access Administration Interface" controls whether users can access the backend of Plotalot. By default, only Super Users have access, but you can grant access to other user groups if you wish.

Regardless of the "Access Administration Interface" setting, only Super Users can save changes to the Plotalot CSS file, import charts, or edit or save DatabasesOptions

Hide Sub-menu Option

There is an option to hide the Plotalot component sub-menu. Depending on the size of your screen and whether you have the main menu expanded or collapsed, you might prefer to hide the sub-menu.

Compact Mode Option

In Joomla 4 and 5 this reduces the font sizes and spacing to fit more information on the admin pages, which is a useful improvement on small laptops.

Background colour

The background colour for the chart area of the chart editor. You will only see this colour behind a chart with a transparent background. You might want to set it the same as the background colour of your website, so that you can get a better idea of how charts will look on your site. The default setting is 'transparent'.

Select Only

Defaults to On, which means that Plotalot only allows SQL queries that start with SELECT or (SELECT. With this setting you can be confident that Plotalot cannot make any changes to your data. If you switch this option off, you can use any valid SQL query, including queries that update data.

Multiple Queries

Defaults to Off, which means that plots can only use a single query. If you enable this option you can use multiple queries, separated by semi-colons. When a plot query uses multiple queries, all except the last query are merely executed. The result set for the chart is obtained from the last query. For example you might want to call a stored procedure before generating your chart values:

```
CALL `my-procedure`; SELECT `col1`, `col2` FROM `my-table`;
```

Fix Nulls

Defaults to On, which means that when Plotalot finds nulls in your sql results, it changes them to a zero (0) for numeric columns, and an empty string ('') for text columns. It also issues a warning in the chart editor. No warning is issued when such a chart is drawn on the front end.

If you want null values to be included in your chart data as actual null values, you can set this option to "No". You might want to do this if you plan to use the Google Charts InterpolateNulls option to guess the value of missing data points, or if you want to see actual gaps in your charts.

If neither of these options is right for you, your query can make use of the SQL IFNULL( ) function to return a specific value in place of a null value.

The Experimental API

If you need features of the Google Charts API that have not yet been released, you can set this option to "On". This should not be used on live sites. You can read more about the Google Charts API release process here:

https://developers.google.com/chart/interactive/docs/release_notes#ReleaseProcess

Specific API Version

If a non-zero value is specified here, the specified Google Charts API is used, overriding the "Experimental API" option.

At the time of writing, the current version of the Google Charts API is 50, and it is causing problems on some sites. If your charts show the error "b.split is not a function" or "Cannot read property 'offset' of undefined", please set this option to 49 to load the previous version of the API.

To revert to using the "current" version of the API, set this option to 0 (zero).

Hide Sample Charts

If you no longer need to see the sample charts in the chart list, use this option to hide them.

Generate Charts on Open

Normally when you click on a chart to edit it, the editor opens with the chart shown. If you have charts with SQL that takes a long time to run, you can switch this option off. Charts are then only shown when you use the Save button.

Locale

For many purposes you can control dates and numeric formats yourself using the features of Plotalot, but in some cases you may need to specify the Locale used by the Google Charts API, as described here:

https://developers.google.com/chart/interactive/docs/basic_load_libs#loadwithlocale

The Locale affects the formatting of values such as currencies, dates, and numbers.

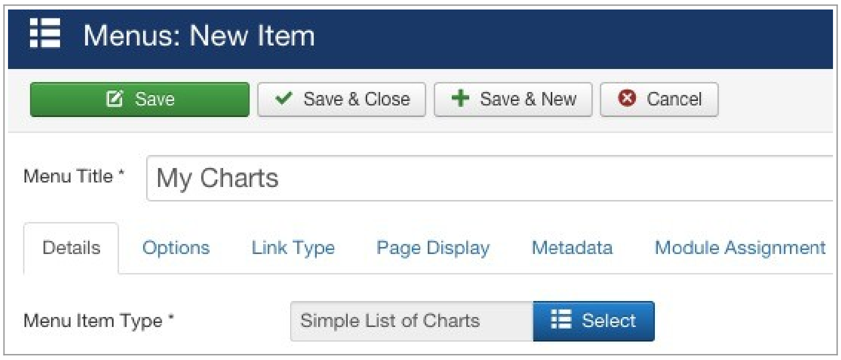
The setting here in the Plotalot Component Options sets the default locale used for all Google charts on the site. You can also set a specific locale for each chart or Google table, which overrides this global setting.

SHOWING CHARTS USING A MENU ITEM

The best way to add charts to your site is to use the **Plotalot Plugin**, which allows you to add as many charts as you like, anywhere in any article. The plugin also performs caching of charts to improve performance.

You can, however, add charts to your site in a simplistic way by using the Component View.

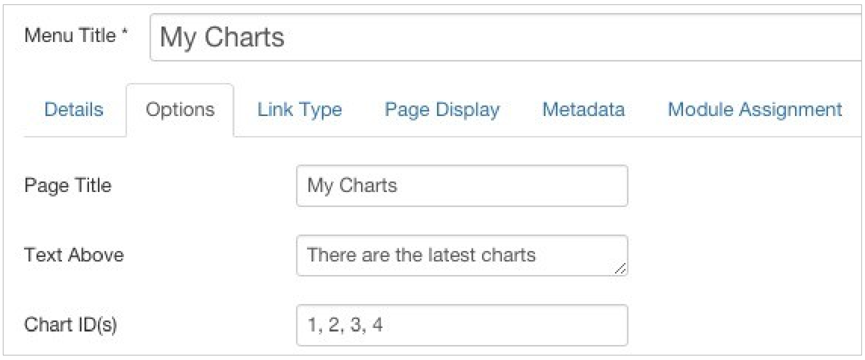
Configure a menu item of type Plotalot, "Simple List of Charts".



Menu Title * My Charts

Menu Item Type * Simple List of Charts

On the Options panel, you can then enter a page title, some text to be displayed above and below your charts, and a list of chart ID’s. You can show more than one chart by supplying a comma delimited list. But there is no way to control the layout or appearance here.



Menu Title * My Charts

Page Title My Charts

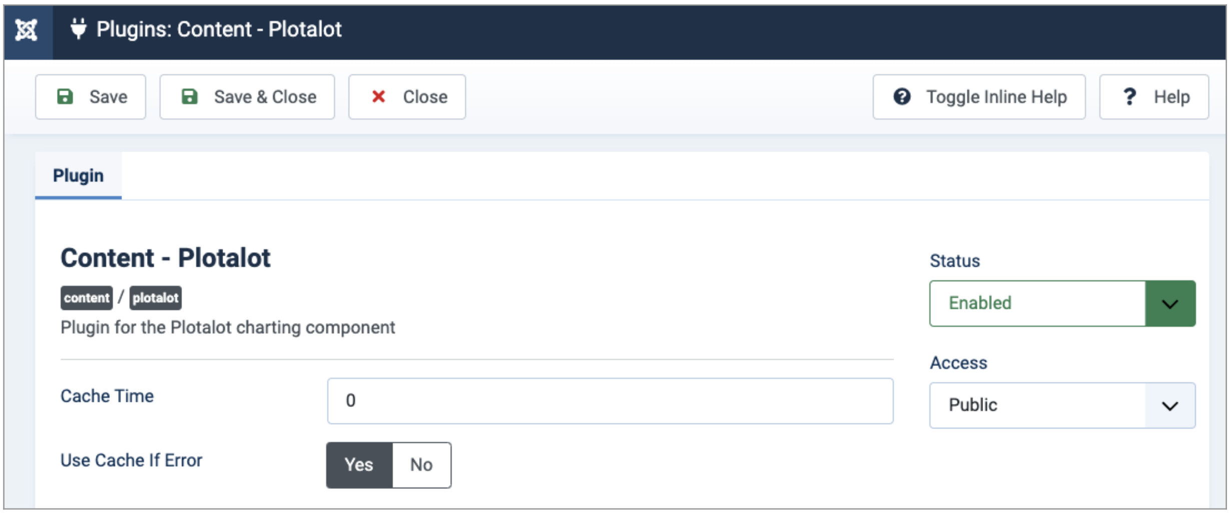
Text Above There are the latest charts

Chart ID(s) 1, 2, 3, 4

THE PLOTALOT PLUGIN

The best way to show charts on your site is to use the Plotalot Plugin and include charts in articles.

Install the Plugin and make sure it is enabled.



Content - Plotalot

content / plotalot

Plugin for the Plotalot charting component

Cache Time: 0

Use Cache If Error: [✓ Yes] No

Status: Enabled

Access: Public

Once you have defined a chart and installed and enabled the plugin, it's easy to add the chart to an article. In its simplest form, the syntax to include the chart with an ID of 1 in an article is simply:

```
{plotalot id="1"}
```

The chart ID must be enclosed in double quotes. Note that Joomla Article Text Filtering and some article editors can interfere with the quotes, causing the chart ID not to be recognised. See the Troubleshooting section for more details.

The plugin cannot display the same chart more than once on the same page. If you need to do that you must make copies of the chart so that each instance has a different chart ID.

Full Width

By default, the plugin draws charts in a containing <div> like this:

```
<div style="display:inline-block;">...</div>
```

This makes the <div> the size of the chart, and allows other content to flow around the chart.

The plugin has an optional parameter, "full_width", which removes the "display:inline-block;" attribute. The <div> now defaults to display:block, and occupies the full width of its container.

```
{plotalot id="1" full_width}
```

Results in this:

```
<div>...</div>
```

If the chart has its width set to blank or zero, the chart will now occupy the width the container, whatever that may be. If the window resizes (for example when a mobile device is rotated), the chart will be re-drawn at the new width. This is a quick and easy way to make charts responsive. A more powerful method is described later.

Auto-Refresh

The "refresh" parameter automatically re-draws charts at regular intervals. The chart data is re-loaded from the server using Ajax, and the chart is re-drawn in-situ without re-drawing the entire page. This works for all charts and Google Tables, but not for Plotalot tables, Responsive tables, or single items.

```
{plotalot id="42" refresh="30"}
```

In this example the chart will re re-drawn using the latest data every 30 seconds.
You can see auto-refresh in action by making a Gauge chart with a query of:

```
SELECT "Time", SECOND(CURRENT_TIME())
```

Adding a Class

The "class" parameter also removes the style="display:inline-block;", and adds a class instead.

```
{plotalot id="1" class="half"}
```

Results in this:

```
<div class="half">...</div>
```

Error Handling

Some kinds of errors can prevent a chart being drawn. You may want to specify what replaces a chart if it cannot be drawn, for example if its source database goes offline, or its data feed fails,. If you do not specify custom error handling the default is for the plugin to report the error like this:

```
{Plotalot Plugin: chart 6 - No rows}
```

This is useful during development but is not ideal for a live site. You have several options.

You can specify an image to be shown in place of the chart. For example:

```
{plotalot id="42" error_img="images/stories/alternate_image.jpg"}
```

Or you can specify some text to appear in place of the chart. For example:

```
{plotalot id="42" error_txt="Sorry, there is no chart data today"}
```

Or you can allow the plugin to use the last cached version of the chart. See "Caching" below.

Caching

The Plotalot Plugin can cache your charts on your server, reducing the amount of work needed to re-produce the chart for the next user who requests it. For charts that use complex queries or charts that don't change very quickly, this is a worthwhile optimisation.

When it produces a chart, Plotalot first has to execute the database query(ies), then analyse the data and produce the Javascript that will draw the chart. The Javascript becomes part of the web page delivered to the browser, and the browser executes the Javascript to render the chart. When the Plotalot Plugin caches a chart, it stores the Javascript in a file on the web server. If the same chart is requested within the allowed cache time, Plotalot retrieves the text file instead of accessing the database and creating the chart from scratch.

The "Use Cache If Error" option determines if the plugin can use an old, expired, cached version of the chart if a new chart cannot be drawn. For example, if the data feed to your database fails, it might be better to show the most recent chart rather than an error image or error message. If set to "Yes", this option takes precedence over the "error_img" and "error_txt" parameters.

You can specify the "Cache Time" and "Use Cache If Error" options globally for all charts in the Plugin Manager, or on the plugin call. Parameters passed on the plugin call take precedence. To enable caching for all charts, set the "Cache Time" in the Plugin Manager to a non-zero number. This is the number of seconds the cached version is valid.

Here's how to specify the options on the plugin call:

```
{plotalot id="42" cache_time="60" use_cache_if_error="yes"}
```

Summary of Plugin Parameters

Parameter	**Notes**
id="nnn"	The chart number to draw. This parameter must always be specified. If the value is non-numeric, it is resolved as a request variable.
cache_time="nnn"	Number of seconds to re-use charts without re-generating them. You can set the default for this parameter in the Joomla Plugin manager
use_cache_if_error="yes"	"yes" means that an expired cached chart can be used if the chart cannot be drawn. You can set the default for this parameter in the Joomla Plugin manager.
full_width	Draws the chart in a <div> without style="display:inline-block;".
error_img="xxx"	Pathname of an image to draw if the chart cannot be drawn.
error_txt="xxx"	Text to be shown if the chart cannot be drawn.
refresh="nnn"	The chart will be re-generated and re-drawn every nnn seconds.
class="xxx"	A CSS class to add to the chart <div> element.
csv="xxx"	Options for CSV data downloads. **csv="my_file_name.csv"** sets the name of the downloaded file. To be recognised as a filename, the name must have the .csv extension. The default filename is "chart_data.csv". **csv="headrow"** adds the column names as the first row of the file. c**sv="sep"** adds an extra first row "sep=," (without the quotes) which forces Microsoft Excel to interpret commas as separators. Do not use this unless your users exclusively use MS Excel as other software will treat this as a data line. Any combination of options can be used by separating them with commas, for example **csv="my_file_name.csv ,headrow,sep"**.
png="xxx"	Options for PNG image downloads. **png="my_file_name.png"** sets the name of the downloaded file. To be recognised as a filename, the name must have the .png extension. The default filename is "chart.png".
P000 - P999	Parameters that can be passed into the chart query, as described in the Advanced Techniques section.

LAYOUT AND POSITIONING

It's up to you to construct the layout of your pages and charts, but here are some ideas.

You can use an inline style to float the chart to the right, so that text flows around it to the left.

```
<div style="float: right; margin-left: 5px;">{plotalot id="2"}</div>
```

You can place two charts side by side using a table.

```
<table>
<tr><td>{plotalot id="1"}</td><td>{plotalot id="2"}</td></tr>
</table>
```

Using tables like this is not considered good practice but is quick and easy for inexperienced users. See the Responsive Charts section for a better solution.

You can place a background image behind a transparent or semi-transparent chart.

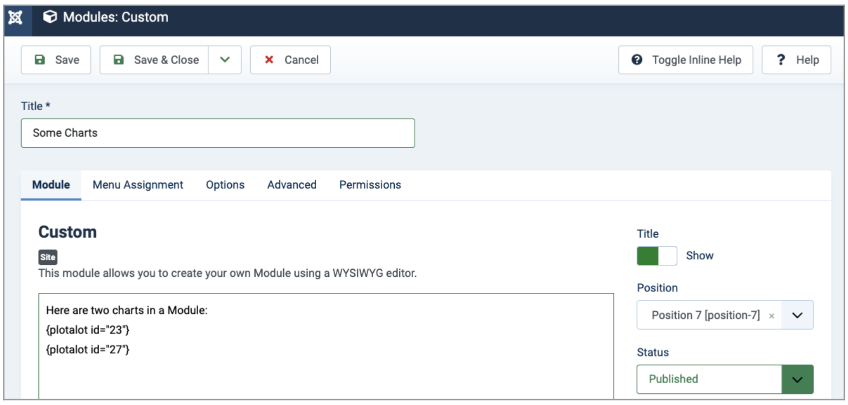
```
<div style="background:transparent url(images/stories/beach.jpg)
      top left no-repeat;">
     {plotalot id="23"}
</div>
```

You can place one chart on top of another. With care, this can be an effective way of showing compound charts, or additional axes.

```
<div>{plotalot id="1"}</div>
<div style="position:relative; top:-180px;">{plotalot id="2"}</div>
```

SHOWING A CHART IN A MODULE

You can show one or more charts in a module by using the Joomla Module Manager to create a new Custom module. In the Custom field, you can put any content you like, including calls to the Plotalot plugin.



Title * Some Charts

Custom

Site

This module allows you to create your own Module using a WYSIWYG editor.

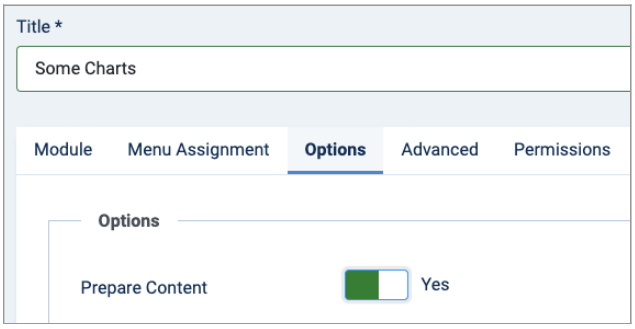
Here are two charts in a Module:
{plotalot id="23"}
{plotalot id="27"}

Title: Show

Position: Position 7 [position-7]

Status: Published

It's very important to enable the "Prepare Content" option (on the Options tab). This tells Joomla to process plugins in the content. Without this option enabled you will just see the unprocessed {plotalot} call.



Title * Some Charts

Options

Prepare Content: Yes

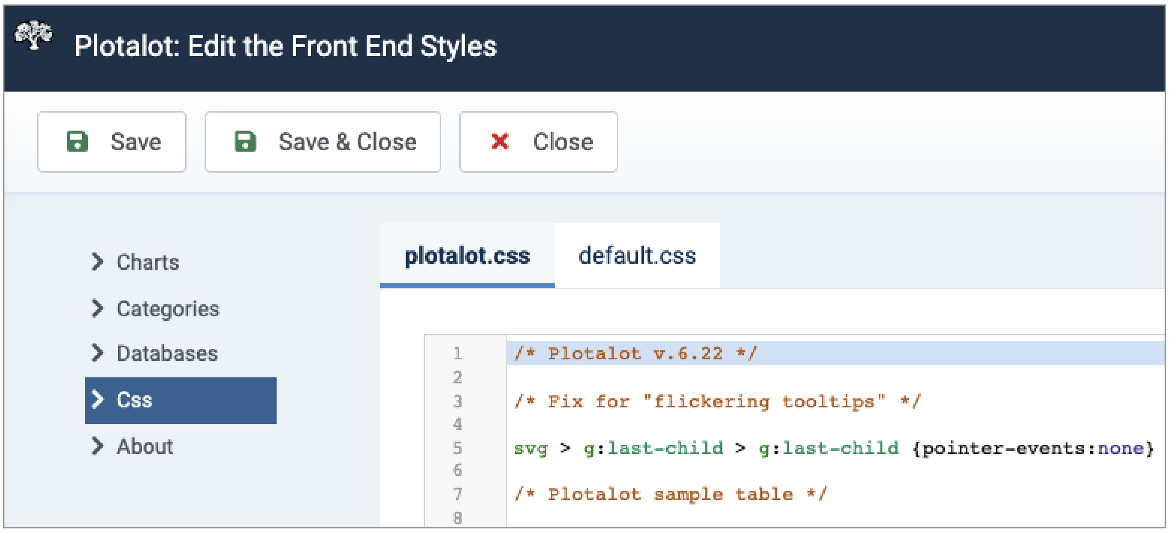
CSS And Responsive Charts

THE PLOTALOT CSS FILE

Whenever Plotalot loads charts on your site, it loads this CSS file, if it exists:

```
<site>/media/com_plotalot/plotalot.css
```

The "CSS" menu item is a quick and easy way to edit this file. You can add CSS rules here to control the styling of your tables, or to control the responsive properties of your charts.



```
/* Plotalot v.6.22 */

/* Fix for "flickering tooltips" */

svg > g:last-child > g:last-child {pointer-events:none}

/* Plotalot sample table */
```

When you update or reinstall Plotalot, the default.css file is updated, but your plotalot.css file is not overwritten. The plotalot.css file is only created (from default.css) if it does not already exist.

The "CSS" menu item includes a tab with the default.css file so that you can easily compare your plotalot.css file with the latest default.css file.

MAKING CHARTS RESPONSIVE

The first step in making charts responsive is to set the width of the chart to blank or 0 (zero). Charts that have a defined width and height will always be drawn at the specified size and will never change their size. Charts with zero width are drawn to match the size of their containing element. We can then use CSS to control the size of the containing element, and its behaviour at different window sizes.

Google charts do not automatically re-size in the way that ordinary images do. In order to be responsive, the browser must be told when to re-draw them. The Plotalot ***Plugin*** includes Javascript to do this. You can see this working on our website:

https://www.lesarbresdesign.info/extensions/plotalot

This dynamic resizing works in all modern browsers, but not in IE8 or below.

The Plotalot Component view does ***not*** include the responsive Javascript. If you use the Component view to display charts of width zero, they will initially be drawn the size of their containing element (which is partially responsive behaviour) - but they will not dynamically resize if the window size changes, for example if a mobile phone or tablet is rotated.

A few templates (for example Gridbox templates from Balbooa) do not allow their grid elements to resize as the browser window resizes. Google charts cannot be made responsive in such templates.

Full Width Charts

The easiest way to make a chart responsive is to set its width to zero and use the plugin's "full_width" option. This will make the chart the full width of the article it appears in, whatever that happens to be at any given time. As the window size changes, the plugin will re-draw the chart. In many cases this is a simple and effective solution.

Normally charts are drawn in a <div> element with a style="display:inline-block" attribute so that the chart merges inline with the flow of the article. But when "full_width" is specified, Plotalot omits the style="display:inline-block" attribute from the <div> element that it draws the chart in. Since the <div> is now display:block, it occupies the full width of its container. As the window size changes, so does the <div> containing the chart, and the plugin re-draws the chart to match it.

Sizing Charts With CSS

If you want to control exactly how your charts resize you can use CSS. Plotalot always draws charts in a <div> element with an "ID" attribute that includes the chart ID, which means you can target them with the CSS # selector:

```
HTML:    <div id="chart_17" style="display:inline-block"></div>
```

```
CSS:     #chart_17 {height:200px; width:350px;}
```

You can define your chart <div> sizes using pixels or percentages, and define their minimum and maximum sizes using the CSS properties "min-width" and "max-width". You can use CSS media queries to control how they respond to different window sizes.

In this example, the chart is initially set to 350 x 200 pixels. But when the screen width is below 750 pixels, the height changes to 150px and the width changes to 100%, meaning that it will occupy the full width of its containing element.

```
#chart_17 {height:200px; width:350px;}
@media screen and (max-width:750px)
{
    #chart_17 {height:150px; width:100%; }
}
```

Two Charts Side-by-Side And Responsive

Let's put two charts side-by-side. This time we'll use the plugin's "class" parameter instead of targeting the charts by ID.

```
{plotalot id="40" class="half"} {plotalot id="42" class="half"}
```

This results in two <div> elements, like this:

```
<div id="chart_40" class="half"></div>
<div id="chart_42" class="half"></div>
```

On a wide enough screen, we want them to sit side-by-side. But on a small screen, we want them to be full width, one below the other. This CSS will do that:

```
.half {display:inline-block; height:200px; width:48%;}
@media screen and (max-width:750px)
{
    .half {height:150px; width:100%;}
}
```

On a wide enough screen, each chart will be a little less than half the width of the page, allowing them to sit side by side. As the screen width reduces they will reduce proportionally, remaining remaining side-by-side until the window falls below 750 pixels, at which point both charts become 100% wide, forcing them to flow one below the other.

For non-trivial layouts, you need to be careful that your article editor does not add unwanted html elements around or between your charts. Some wysiwyg editors add lots of <p> and <div> elements that can upset your responsive layouts. For this kind of work, we usually set the Joomla article editor to CodeMirror or None.

Multiple Responsive Charts On The Same Page

You can have multiple responsive charts on the same page, but only if they are shown by a single article or module. If you combine charts from more than one article (or module) on the same page, only the charts in one of the articles will be responsive.

STYLING THE PNG AND CSV BUTTONS

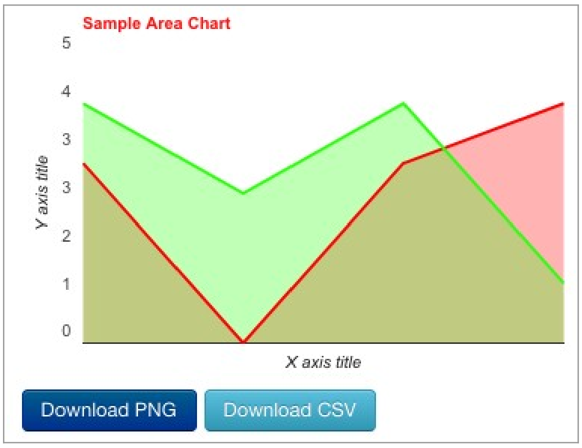
The PNG and CSV links can include HTML, so you can specify the links as buttons with CSS classes. For example, if your template loads bootstrap.css, you can use the "btn" classes, like this:

```
<button class="btn btn-primary">Download PNG</button>
```

and

```
<button class="btn btn-info">Download CSV</button>
```

The buttons are wrapped in a <div> of class "pl_pbuttons". The PNG button is wrapped in a <div> of class "pl_plink", and the CSV button is wrapped in a <div> of class "pl_clink".



This section is intended only for advanced users with relevant Joomla and web development experience. Non-technical users are unlikely to succeed in using the techniques described in this section. Since discussions about these techniques can be complex and time-consuming, support for the features and techniques described in this section is only available for users who have purchased Plotalot Option 3.

PLOTALOT VARIABLES

You can make use of the following variables in any text or query.

%%J_USER_ID%%	The id of the currently logged on user
%%J_USER_NAME%%	The name of the current user
%%J_USER_USERNAME%%	The login/screen name of the current user
%%J_USER_EMAIL%%	The email address of the current user
%%J_ROOT_URI%%	The URI of the current site, e.g. http://www.mysite.com/
%%J_ROOT_PATH%%	The file system path of the current site, e.g. G:\xampp\htdocs\test
%%J_CURRENT_DATE%%	The current date as YYYY-MM-DD, retrieved using Joomla's date functions.
%%Pn%%	Any number of Plugin Variables. These are defined by an upper case 'P' followed by a number. For example P0, P1 … P9, P10, P11, etc.
%%Pn=default value%%	Plugin variables can have default values.

Plotalot resolves variables before executing queries. For example:

```
SELECT `name`, `username` FROM `#__users` WHERE `id` = '%%J_USER_ID%%'
```

In the back end, this will retrieve the name and username for the currently logged on administrator. In the front end it retrieves the name and username for the currently logged on user.

PLUGIN VARIABLES

Plugin Variables allow you to pass variables on the call to the Plotalot plugin. For example:

```
SELECT UNIX_TIMESTAMP(`date`), `%%P2%%` FROM `#__daily_summary`
     WHERE `date` BETWEEN '%%P1%%-01-01' AND '%%P1%%-12-31'
```

This query uses two variables. P1 is the year number used as part of the where clause. P2 is the name of second column to retrieve.

On the plugin call, you could populate these variables like this:

```
{plotalot id="68" P1="2009" P2="average_value"}
```

Plugin Variables only apply when using the Plugin.

You cannot currently use Plugin Variables with Auto-Refresh.

DEFAULT VARIABLE VALUES

Plugin Variables can have default values. The default value is used if the plugin cannot find a value for the variable. For example:

```
SELECT UNIX_TIMESTAMP(`date`), `%%P2=average_value%%` FROM `#__daily_summary`
      WHERE `date` BETWEEN '%%P1=2007%%-01-01' AND '%%P1=2007%%-12-31'
```

Default values are also useful for testing your chart in the back end, where Plugin Variables do not have values. Default values can only be literal strings. They cannot be variables. However, you can work around this limitation in SQL, for example:

```
      IF("%%P1%%" = "", "%%J_USER_USERNAME%%", "%%P1%%")
```

VARIABLES IN PLAIN TEXT

Variables don't have to be used in sql queries. They can also be used in plain text. For example, in the above example, the chart title could be:

```
      %%P2=Average%% Temperatures for %%P1=2016%%
```

The default chart title would then be drawn as "Average Temperatures for 2016". But the title would change if P1 or P2 resolved to different values.

USING GET AND POST DATA

You can tell the Plotalot plugin to retrieve data values from Get or Post variables, by specifying their names with a leading underscore. For example:

```
      {plotalot id="nn" P1="_temperature"}
```

Passes the value of the Get or Post variable "temperature" as the value of plugin parameter P1.

Get Variable Example

Make a gauge chart with this SQL:

```
      SELECT "Temperature", %%P1=0%%
```

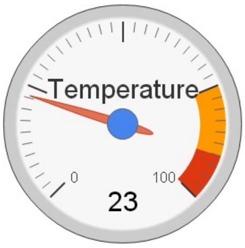
P1 is a plugin variable with a default value of zero. In the chart editor, the gauge shows zero. Now make an article with this plugin call:

```
      {plotalot id="nn" P1="_temperature"}
```

Where nn is the id of the gauge chart. Make a menu item to access the article, and look at the article on the front end of your site. The gauge still shows zero. Now add the temperature parameter to the article URL:

```
      http://mysite.com/index.php/plotalot-test?temperature=23
```

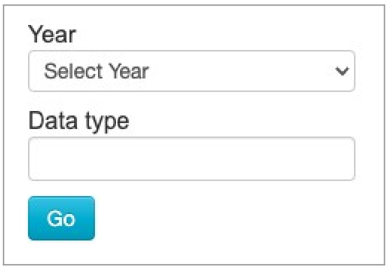
The plugin retrieves the "temperature" variable from the URL, and sets it as the value of P1. The SQL statement now returns 23 as the second column of the query, and the gauge shows this value.



Post Variable Example

It is possible to construct forms in Joomla articles, although you do need to disable your wysiwyg editor to do so. Build an HTML form in an article, like this:

```
<form action="" method="get">Year<br />
<select name="year" size="1">
   <option value="2019">2019</option>
   <option value="2020">2020</option>
   <option value="2021">2021</option>
   <option value="2022">2022</option>
</select>
<br />Data type<br /><input type="text" name="column_name" size="15" >
<br /><input type="submit" value="Go" class="btn btn-info">
</form>
{plotalot id="68" P1="_year" P2="_column_name"}
```



The empty <form> action submits the form back to the same URL, so in this case, the article shows the form and the resulting chart. When you click the "Go" button, the form submits the form with the variables "year" and "column_name" set according to the user's input.

The plugin draws chart 68 with variable P1 set to the chosen year, and variable P2 set to the entered column name. In the chart SQL you can use these two values to show a chart that reacts to the user's selections.

Arrays in Get and Post Data

If you wanted the user to be able to select multiple years, you could use a "multiple select" in the previous example:

```
<select name="year[]" size="4" multiple="multiple">
   <option value="2019">2019</option>
   <option value="2020">2020</option>
   <option value="2021">2021</option>
   <option value="2022">2022</option>
</select>
```

The user can now use the Ctrl and/or Shift keys to select multiple years. The multiple select element sends the selected years back to the server as an array:

```
      [year] => Array (
            [0] => 2020
            [1] => 2021
            [2] => 2022 )
```

The Plotalot Plugin converts Get and Post arrays to comma-delimited strings, so the array in our example becomes: '2020', '2021', '2021' - which is ideal for use in a SQL "IN" clause:

```
      SELECT * FROM `#__some_table` WHERE `year` IN (%%P1%%)
```

Another Example

Here's the same example using checkboxes instead of a multiple select:

```
<form action="" method="get">Year:
2019 <input name="year[]" type="checkbox" value="2019">
2020 <input name="year[]" type="checkbox" value="2020">
2021 <input name="year[]" type="checkbox" value="2021">
2022 <input name="year[]" type="checkbox" value="2022">
</select>
<br />Data type: <input type="text" name="column_name" size="15" >
<input type="submit" value="Go">
{plotalot id="68" P1="_year" P2="_column_name"}
```

PASSING PHP VARIABLES FROM SOURCERER

Sourcerer is a third party Plugin that enables you to include PHP code in an article. You could calculate a value in PHP and then pass the result to a Plotalot chart by setting a variable that the Plotalot Plugin can read. For example:

```
{source}
<?php
$degreef = 123;
$degreec = round(($degreef - 32) * 5/9, 1);
$jinput = JFactory::getApplication()->input;
$jinput->set('temperature', $degreec);
?>
{/source}
{plotalot id="9" P1="_temperature"}
```

ASSIGNING THE RESULT OF A SINGLE ITEM

It is also possible to retrieve the result of a Plotalot plugin call in PHP code in an article. When Joomla renders an article, any plugins replace their calling text with their output text. For charts, the output text is the HTML <div> element that the chart will be drawn in. For Plotalot tables, it's the <table> HTML. You are unlikely to want to manipulate either of those things in PHP in an article. But for a single item, the result is the output of the database query defined for the single item, and in some cases it can be useful to manipulate this within the article.

The trick here is to ensure that the Plotalot plugin values are resolved before the PHP code in the article is executed. You can't do that with Sourcerer because it is a System Plugin, but you can do it with DirectPHP, which is a Content Plugin. You can use the ordering column of the Joomla Plugin Manager to ensure that Plotalot runs before DirectPHP.

In the following example, DirectPHP will not "see" the {plotalot ...} calls because they will have been resolved and replaced with their result values before DirectPHP is called. So in the following example, DirectPHP will run with the output values of the two Plotalot calls, and will execute the PHP code that adds them together and echoes the result:

```
<?php
$value1 = {plotalot id="1"};
$value2 = {plotalot id="2"};
$value3 = $value1 + $value2;
echo "Result is $value3";
?>
```

PLUGIN VARIABLE CHART ID

When you call the plugin, you normally specify a Chart ID like this:

```
{plotalot id="68"}
```

If you specify a non-numeric Chart ID, the plugin retrieves the value as a Get or Post variable. For example:

```
{plotalot id="chart_id"}
```

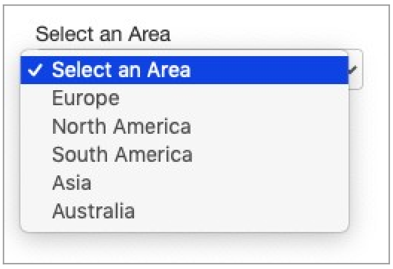
Retrieves the value of the "chart_id" variable. You could pass the "chart_id" parameter on a URL, like this:

```
http://www.mysite.com/my-charts?chart_id=68
```

Selecting a Chart From a List

You can use a variable chart ID to create a select list of charts in an article:

```
<form action="" method="get">Select an Area<br />
<select name="cid" size="1" onchange="this.form.submit()">
   <option value="0">Select an Area</option>
   <option value="38">Europe</option>
   <option value="39">North America</option>
   <option value="40">South America</option>
   <option value="41">Asia</option>
   <option value="42">Australia</option>
</select>
</form>
{plotalot id="cid" error_txt=" "}
```


Select an Area

✓ Select an Area
Europe
North America
South America
Asia
Australia

There are a few things to explain here:

- The "onchange" attribute submits the form automatically when a selection is made.
- The select list sends back a Get parameter, "cid", containing the selected chart ID.
- The chart ID in the plugin call is "cid", so the plugin will use the value of "cid" as the chart ID.
- The first time the article is opened, "cid" will be empty, so to avoid an error, we added the error_txt parameter with a single space, so that nothing is shown.

LOADING CHARTS ASYNCHRONOUSLY

If you have several charts on one page, and they take a long time to load, you could consider showing them in HTML iframes. Iframes load asynchronously, using several active connections to the server at the same time. For example, your article could include iframes like this:

```
<h2>Please wait for the charts to load</h2>
<iframe src="http://mysite.com/index.php?view=article&id=96&tmpl=component"
style="width:100%; height:500px;"></iframe>
<iframe src="http://mysite.com/index.php?view=article&id=97&tmpl=component"
style="width:100%; height:500px;"></iframe>
```

The tmpl=component parameter tells Joomla to return the component output (i.e. in this case, the article) without the rest of the page content.

To avoid having to create a separate article for each chart, you could use the Variable Chart ID as described above. Notice that in the first example we loaded two different articles (96 and 97). Here we are re-using the same article (96), and adding the chart_id variable:

```
<h2>Please wait for the charts to load</h2>
<iframe src="http://mysite.com/index.php?view=article&id=96&tmpl=component
     &chart_id=68" style="width:100%; height:500px;"></iframe>
<iframe src="http://mysite.com/index.php?view=article&id=96&tmpl=component
     &chart_id=69" style="width:100%; height:500px;"></iframe>
```

CREATING A PRINTER FRIENDLY BUTTON

The site https://www.printfriendly.com/button can help you to generate a button that generates a printer friendly version of a web page. It works quite well, except that it incorrectly shows a hidden data table that is generated for accessible screen-readers. You need to hide this data table with a little extra CSS:

```
      [aria-label~="tabular"] {display:none}
```

EXTRA JAVASCRIPT

If you know Javascript there are many ways to enhance Google Charts. The Javascript you enter in the Extra Javascript field is added to the chart script just after the data table has been built, just before the draw function is called. The Google Charts API provides many functions for manipulating the chart data and the chart itself. You can access the data table and the chart object. For example, if the chart ID is 9, the chart is referenced as "window.plotalot_chart_9", and the data table is referenced as "window.plotalot_chart_9_data".

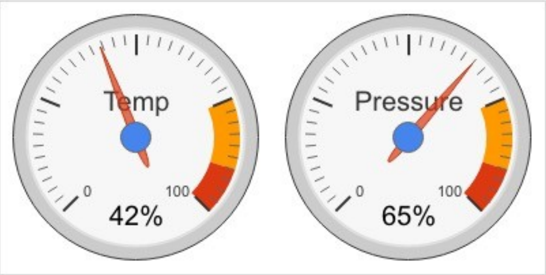
Adding a Formatter Function

You can add a formatter function to assign a custom format to a data column. In this example we will add a '%' suffix to the value of the Sample Gauges chart, using the NumberFormat function documented here:

https://developers.google.com/chart/interactive/docs/reference#numberformatter

Assuming the chart ID is 9, here's the code you would add to the Extra Javascript field:

```
var formatter = new google.visualization.NumberFormat({suffix:'%',pattern:'#'});
formatter.format(window.plotalot_chart_9_data, 1);
```



Adding an Event Handler

In this example we will use Extra Javascript to add an event handler to the Sample Organisation Chart. The event handler copies the selected name to a global variable for use by some external Javascript.

```
google.visualization.events.addListener(window.plotalot_chart_20, 'select',
function()
   {
   if (window.plotalot_chart_20.getSelection().length)
      {
      var item = window.plotalot_chart_20.getSelection()[0];
      var row = item.row;
      window.selected_name = window.plotalot_chart_20_data.getValue(row, 0);
      alert(window.selected_name);
      }
   }
);
```

How Plotalot Works

Plotalot uses the Google Charts API, which is a library of Javascript functions. The library renders charts using HTML5 Scalable Vector Graphics (SVG). Vector Markup Language (VML) is used for older versions of Internet Explorer. The library works with all modern browsers including those on iPhones, iPads and Android. Charts are rendered on the client computer, using the browser's Javascript engine. Google's servers are not involved in rendering the charts, and the chart data is not sent to Google or any other server. Google simply serves the Google Charts Javascript library.

Plotalot retrieves chart data from the database by executing the SQL select statement(s) configured for the chart. It then analyses the data to determine the data types and value ranges, and builds the Javascript data structure in the correct format for Google Charts API and the specific chart. For charts with multiple plots, the datasets must be merged into a single dataset, and in some cases re-sorted. Values and strings are formatted, encoded and escaped as required. Plotalot then produces the rest of the Javascript needed to call the Charts API, based on the options selected. The Javascript is added to the Joomla page being generated, and is executed by the browser on the client computer, with the chart eventually being rendered in the chosen <div> element.

Part of Plotalot's analysis decides on the number of data points to include in the Javascript. There is no point sending more data points to the browser than the chart has pixels. Doing so would be unnecessarily slow, and might even crash the browser. For large datasets, Plotalot samples the data to provide accurate charts without creating unduly large scripts. The sampling process does mean that data "spikes" could be missing from the data sent to the browser, but there is really no choice, and it's likely that spikes of less than a pixel would not be visible in any case.

You can examine the Javascript produced by Plotalot by looking in the Script tab in the chart area, or by viewing the source of a web page containing Plotalot charts.

Language Translations

All of the fixed text displayed in Plotalot uses the standard Joomla translation system, with language strings held in a language file. Plotalot only needs a language file for the back end as there are no texts that need to be translated for the front end.

The main Plotalot language file is (for English):

```
<site>/administrator/components/com_plotalot/language/en-GB/en-GB.com_plotalot.ini
```

Plotalot has language files for several languages. All of them are installed when you install the component.

To make a new language file, make a copy of one of the existing language files to a new file of the correct name for your language, then translate the text on the right hand side of the = signs. For example:

`en-GB.com_plotalot.ini`	`NEW_CHART="New chart"`
`fr-FR.com_plotalot.ini`	`NEW_CHART="Nouvelle graphique"`
`it-IT.com_plotalot.ini`	`NEW_CHART="Nuova grafico"`

The format is very critical and any mistakes may result in the language file being ignored by Joomla.

- The double quotes around the translation text are important. You cannot use double quotes inside the text.
- Be careful to preserve any %s markers. These mark the place where text or numeric values are inserted when the text is used.
- Use a text editor to edit the file, not a word processor.
- Make sure you save the file with UTF-8 encoding otherwise accented characters will not be displayed properly.

When you copy the new file to the correct directory on your website, it becomes active immediately. There is no need to reinstall the component.

If your language modifications are not working as you expect, Joomla has a language debugging system which is very easy to use. Go to Global Configuration, System, and enable both debug settings. Then, at the bottom of every page of your site, you will see a lot of information about which language files are loaded, and details of any errors.

If you create a new language file, or update or correct an existing file, please send the file to us so that we can include it in the product. All of our translations have been donated by other users, and most require updating because with each release, we add new features, which require new language strings. If you update language files, please send them to us so that we can include them in the next release of Plotalot and save other users from having to duplicate your work.

PLUGIN DOES NOT SHOW CHART

Please check that the Plotalot Plugin is installed, and that it is enabled in the Joomla Plugin Manager. The Plotalot Plugin must be installed separately from the component and then enabled in the Plugin Manager. There is no free version of the plugin, it is a paid download.

If the Plugin is installed and enabled, but still not working properly, view your article in HTML mode. Some wysiwyg editors make a real mess of things. This is a real example:

```
{plotalot  id="39&lt;br /&gt;}"
```

If you can't get the article to save cleanly, you might also need to disable Article Text Filtering in Joomla Global Configuration.

PLUGIN VARIABLES ARE NOT RESOLVED

Again, check your article in HTML mode. Here's another example from a support case. In wysiwyg mode, the plugin call looked like this:

```
{plotalot id="23"  P1="_product" }
```

But in html mode it looked like this:

```
{plotalot id="23"  P1="_product</span><span style="font-size: 12.16px;">" }
```

So Plotalot was looking for a variable named "product</span><span style=".

PLUGIN DOES NOT WORK IN A CUSTOM HTML MODULE

Check that you enabled the Prepare Content option in the Module options. Unless this option is enabled, the module will not resolve plugins.

ONLY ONE CHART SHOWING

Some types of script optimisation techniques can prevent multiple charts being drawn on a page. The google.charts <script> tags must not be combined into a single <script> tag or file. In the case of the JCH Optimize plugin, changing from "Optimum" to "Average" fixes the problem.

ERRORS WHEN SAVING CHARTS

Errors can occur when saving some of the sample charts if your server has the Apache mod_security module configured, with protection against SQL injection attacks. A common mod_security rule returns a 403 (access denied) error if a request includes a SQL "UNION" statement, because UNION can potentially be used as part of a SQL injection attack. See this page for more details:

http://stackoverflow.com/questions/19448257/403-forbidden-if-the-word-union-is-found-in-url

The error is normally a 403 but we have also seen cases where a 404 error was generated.

To confirm whether this is the problem you are facing, try saving a chart that does not include a SQL UNION statement. If that works, then the problem is probably the UNION mod_security rule. In most real charts, you won't need to use UNION, but if you do, you will need to ask your web host to remove that mod_security rule for your account.

Other mod_security rules can generate errors when saving charts, even without UNION statements. A good way to confirm the problem is to blank out the SQL query and click the Close button. This will not save the chart, but does still transmit all the form data back to the server, so mod_security can still generate an error. If the error goes away without the SQL, the problem is almost certainly caused by a mod_security rule.

Some cPanels allow you to switch mod_security on and off, even if they don't give you access to the rule configuration.

CSV DOWNLOAD NOT WORKING

The CSV download feature relies on the 'download' attribute of the HTML5 <a> element, which is not supported by older browsers. The relevant browser versions are listed here:

> https://caniuse.com/#search=download

CHARTS ARE NOT RESPONSIVE

Charts are only responsive when they are drawn by the Plotalot plugin.

Make sure that the width of your chart is set to blank or 0 (zero). This triggers the responsive behaviour in the plugin.

Responsive charts need a responsive template. Make sure that the <div> element containing the chart is able to change size as the screen size changes. Some templates may prevent this. Use the Chrome Developer Tools or the Firefox Web Developer Inspector to inspect the properties of the <div> containing the chart.

You can temporarily replace the call to the Plotalot plugin with a test container so that you can test the behaviour of your responsive CSS:

```
<div id="chart_21" style="border:2px solid red;">Test</div>
```

Check for fixed width containers in your template. One user had this:

```
<div style="margin:auto; width:600px;">
. . .
{plotalot id="1"}
</div>
```

Check that a Javascript error is not preventing the chart resizing Javascript from running. Use the Chrome Developer Tools Console or the Firefox Web Developer Console to check for Javascript errors on the page.

LANGUAGE TRANSLATION PROBLEMS

Plotalot comes with several language translations contributed by users. Look in the component language directory to see the language files that are installed:

```
<site>/administrator/components/com_plotalot/language
```

Joomla has a language debugging system which is very easy to use. Go to Global Configuration, System, and enable **both** debug settings. Then, at the bottom of every page of your site, you will see a lot of information about which language files are loaded, and details of any errors.

If you create or update language files, please send them to us so that we can include them in the next release of Plotalot.

HORIZONTAL DATE AXIS HAS THE WRONG NUMBER OF LABELS

There has been a long history of problems with date axes in the Google Charts API. The situation is now much better than it was although you may still see occasional anomalies. The discussion thread is here:

https://github.com/google/google-visualization-issues/issues/908

WHY IS THERE NO LINE ALONG THE Y AXIS OF BAR CHARTS?

The line along the X axis is really marking the zero value of the Y axis. The X axis cannot have a zero value because it is made up of strings. It's just the way the Google Charts API works.

INTEGRATING WITH OTHER EXTENSIONS

Sometimes people ask, "How do I integrate Plotalot with extension XYZ", or "How do I draw graphs from data entered by extension ABC"? The answer is … we don't know! There are thousands of extensions for Joomla and we can't give advice about the ones we didn't write. You need to fully investigate and understand the capabilities of the two components you want to integrate, and find a way to make it happen.

If you find a good solution that you think would be a useful addition to this document, please tell us about it and we will consider adding it to this guide to save other people time and effort.

If you think Plotalot needs some kind of new interface or data transferral mechanism that would make it a more useful product, you can propose a technical solution to us and we will be happy to consider it.

REPORTING GOOGLE CHARTS ISSUES

If you discover a bug or a problem with a chart, or you would like to discuss issues at the Javascript level, you can visit the Google Charts forum, here:

https://groups.google.com/forum/#!forum/google-visualization-api

PROBLEM WITH THE FALANG LANGUAGE MANAGER

The Falang language manager intercepts all database queries made on the front end of a site in order to retrieve different versions of content for different languages. It works well for most "normal" database queries, but it does not work properly for some queries that may be used for charts. For example, with Falang installed, many of the sample charts in Plotalot will work perfectly in the back end of Joomla, but in the front end they return errors such as "This chart requires 2 columns".

The solution is to create a Plotalot Database object for the current site, and set your charts to use this database instead of the default of "This site". Plotalot will then make a separate connection to the database, bypassing Falang. You can give this Plotalot Database any name you like, for example "Direct connect". You need to specify the host, database name, user name and password. Use the test button to make sure you got everything correct. Then select this new Database for any charts that are not working properly in the front end.

ERROR "GETIMAGEURI IS NOT A FUNCTION"

This error can occur if you configure a download button for a chart type that does not yet support conversion to .png images. If you see the error "getImageURI is not a function", the chart type does not yet support this feature.

CANNOT WORK OFFLINE

The Google Charts API does not work when the client computer is not connected to the internet. See this page for more information:

http://stackoverflow.com/questions/6124930/google-chart-api-is-it-possible-to-download-the-js-and-work-on-it-offline

B.SPLIT IS NOT A FUNCTION

CANNOT READ PROPERTY 'OFFSET' OF UNDEFINED

Since the release of Google Charts API version 50 on the 11th of April 2021, a small number of users have reported that their charts show the error "b.split is not a function", and two users have reported the error "Cannot read property 'offset' of undefined". The errors occur in multiple chart types but only on a small number of sites.

These errors are not caused by Plotalot. The same errors occur on multiple sites using Google Charts without Plotalot.

This problem is caused by the presence of the MooTools Javascript library as explained here:

> https://groups.google.com/g/google-visualization-api/c/ys2YmOXaq8k

The quick fix is to use the latest version of Plotalot and set the Component Option, "Specific API Version" to 49. This will fix the problem, but it's not a desirable long term solution, since it does not allow your charts to progress to newer versions of the Google Charts API, which continuously introduce enhancements and browser fixes.

The long term solution is to eliminate MooTools from your site. In some cases you might find that you can do this by changing settings in your template or extensions. In some cases you might have to replace extensions with different ones. It will be time well spent. We see too many Joomla sites loading too many Javascript libraries. And it will be good preparation for Joomla 4, which does not support MooTools. Our extensions dropped all usage of MooTools several years ago.

A much less desirable solution may be to upgrade to MooTools 1.6, which does not cause any problem with Google Charts (Joomla 3 ships with MooTools 1.4.5). This is not an easy job, and it might well break the extensions that use MooTools. We don't recommend this approach.

Help And Support

If Plotalot works well for you, please help us by posting a review at the Joomla Extensions Directory:

http://extensions.joomla.org/extensions/financial/graphs-and-charts/11717

If you like Plotalot, please buy the paid version, which also gets you the plugin:

http://www.lesarbresdesign.info/extensions/plotalot/downloads

Please don't ask for help with SQL queries. It takes time to learn SQL, and it takes time to learn Plotalot, but that's no excuse for asking us to do your work for you.

If you find a real bug or problem, we do want to hear about it, and we will try to help, so please contact us, giving us as much detail as you can. Often, it helps to send us a trace as described in the Troubleshooting section.

Please make initial contact using the form on our website - the magic word is "contact":

http://www.lesarbresdesign.info/contact-us

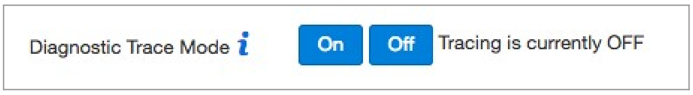
You can find a detailed version history for Plotalot, and check whether you have the latest version, here:

http://www.lesarbresdesign.info/version-history/plotalot

Appendix 1: Trace Mode

Plotalot has a built-in trace feature, which is very useful for resolving complex problems. If you contact us with a problem, we will sometimes ask you to send us a trace. Of course there is nothing to stop you examining the trace yourself before contacting us. You might be able to spot the cause of the problem yourself. The trace is especially useful for understanding how variables are resolved.

To obtain a trace, in the administration interface, go to the Plotalot About page. At the bottom of the page are two buttons. Click the On button. A link to the trace file is displayed, but don't download the file yet.

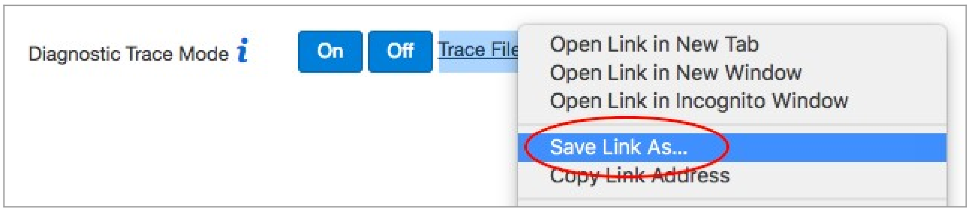


Tracing is currently OFF

Now make Plotalot draw the chart you want to trace. You can use the front end, or the back end, or the plugin. If you use the plugin, make sure Plotalot caching is disabled.

Make sure you do not draw any charts after the one you need help with, because each new chart starts a new trace file.

Now go back to the Help and Support page, right click on the "Trace File" link, and save it to your computer.



Once you have downloaded the trace file, click the Off button to switch tracing off. The trace file will be deleted. Active trace files might be accessible over the web and might contain sensitive information, so it's important to switch tracing off after use.

The trace file contains all of the data retrieved by the query. You can edit the trace file to remove or modify any sensitive data before sending it to us.

Please send trace files to us as attachments. Please don't paste the contents into an email as it obscures the conversation.