How to format your chart and axis titles in ggplot2

sharpsightlabs.com/blog/format-titles-and-axes-in-ggplot2

Once you know how to create simple plots you'll want to learn how to design more sophisticated plots.

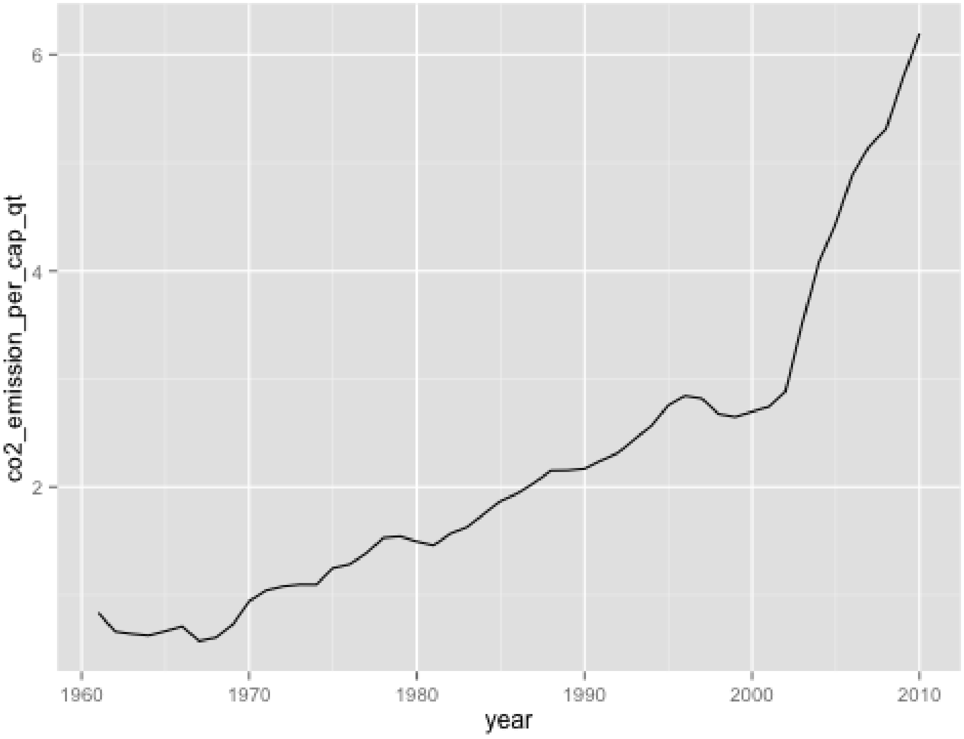
A large part of being able to design sophisticated plots is having control over the "non-data elements" of the plot, such as the plot title and axis titles.You want to be able to format those and polish them for publication and presentation.

In this tutorial, I'll show you how to add plot titles and axis titles to your chart. I'll also show you how to format them. This will serve as an introduction to the general topic of formatting (i.e., theming) within ggplot2.

First, let's load a dataset that we can work with. We'll also quickly plot the data to display the data visually, prior to formatting.

```
df.china_co2 <- read.csv(url("https://vrzkj25a871bpq7t1ugcgmn9-wpengine.netdna-ssl.com/wp-content/uploads/2015/01/china_co2_1961_to_2010_data.txt"))

ggplot(data=df.china_co2, aes(x=year,y=co2_emission_per_cap_qt)) +
  geom_line()
```



This is just very simple line chart. No formatting. Everything here just takes on default values.

Notice that there is no chart title and the axis titles are just the variable names. By default, ggplot() simply uses your variable names as the axis titles.

When you're doing data exploration (and not at the stage where your charts need to be polished) the defaults are fine.

Actually, I recommend that you *don't bother* with chart formatting while you're doing data exploration (assuming that in this stage, you're not showing it to executives, partners, customers, etc). When you're just exploring your data, keep it simple.

But later, you'll need to polish your charts. Formatting matters. *Design matters.* Like it or not, the human mind likes beautiful things. If you can learn to make beautiful charts – charts that appeal to the human mind – your analyses will be taken more seriously. (And if you're really good at making beautiful visualizations, *you* will be much more visible to important people. Trust me on this.)

Ok, let's go back to the code. Now that we've seen an unformatted chart (without any title or axis labels), let's add some titles. We'll add *unformatted* titles first and then format them.

How to add a chart title and axis titles

Adding a chart title and axis titles is relatively easy.

[table]
What you want to do[attr style="width:200px"], function, notes
Add chart title, ggtitle(),""
Add axis titles, labs(), the "x=" and "y=" parameters control the x-axis title and y-axis title respectively
[/table]
You add a chart title with the ggtitle() function.

You add axis titles with the labs() function. Note that the labs() function has an "x=" parameter and a "y=" parameter. Those do exactly what they look like: they add the x-axis title and y-axis title, respectively.

```
ggplot(data=df.china_co2, aes(x=year, y=co2_emission_per_cap_qt)) +
  geom_line() +
  ggtitle("China C02 Emissions") +
  labs(x="Year",y="C02 Emissions")
```

Keep in mind though that the formatting of the plot title and axis titles is controlled outside of the ggtitle() and labs() functions.

You control the title formatting using the theme() function.

The theme() function: formatting in ggplot2

In ggplot2, formatting of "non data elements" is performed with the theme() function.

Think of the theme() function as a formatting system. It doesn't change the data or the geometric objects of the plot. Rather, it changes the "look" and "style" of the plot.

Almost any "non-data element" you want to format can be changed using the theme() function. You just need to call the function by specifying which plot element you want to modify.

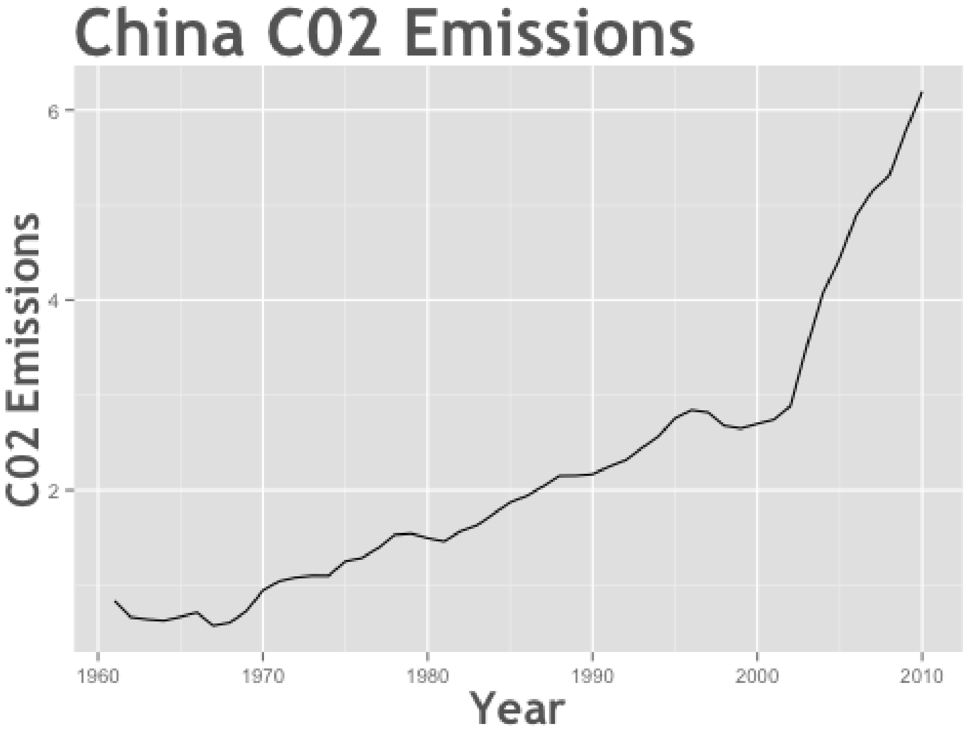
I'll show you an example first, and then we'll use that example as a case that we can break down to understand how the theme() function works.

How to format your chart title and axis titles

As the instructive example, I'll show you how to format the plot title and axis titles that we added to our line chart.

Here we'll call the theme() function to format our plot and axis titles.

```
#-------------------
# FORMATTED TITLES
# - plot.title
# - axis.title
#-------------------
ggplot(data=df.china_co2, aes(x=year, y=co2_emission_per_cap_qt)) +
  geom_line() +
  ggtitle("China C02 Emissions") +
  labs(x="Year",y="C02 Emissions") +
  theme(plot.title = element_text(family = "Trebuchet MS", color="#666666",
face="bold", size=32, hjust=0)) +
  theme(axis.title = element_text(family = "Trebuchet MS", color="#666666",
face="bold", size=22))
```

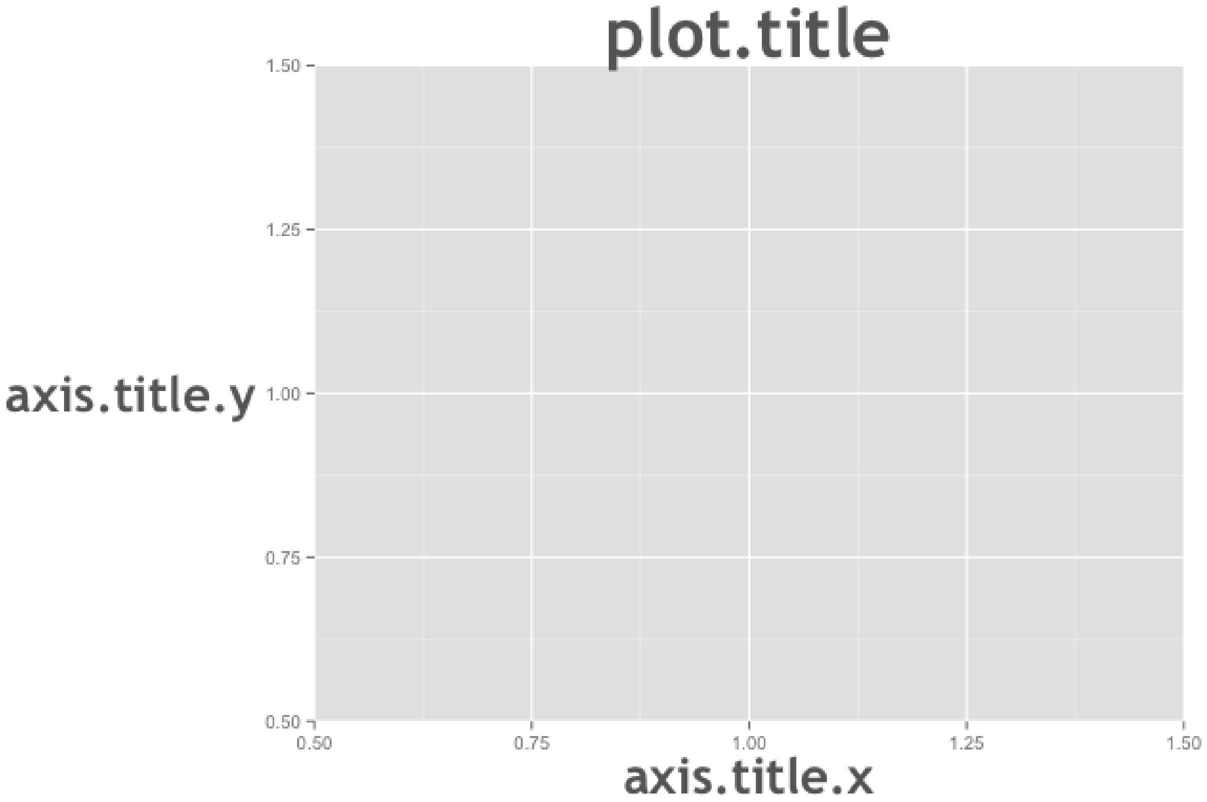


In the above code, we've called the theme() function twice: once for plot.title and once for axis.title.

plot.title and axis.title are “theme elements.” (Notice in the below table that you can modify the x and y axes individually.)

[table width=”500px”]
Theme element, What it refers to
plot.title, the title of the plot
axis.title, both axis titles
axis.title.x,the x-axis title
axis.title.y,the y-axis title
[/table]

And here’s what those “theme elements” actually refer to, visually.



The first argument of the theme() function is the theme element that you want to format.

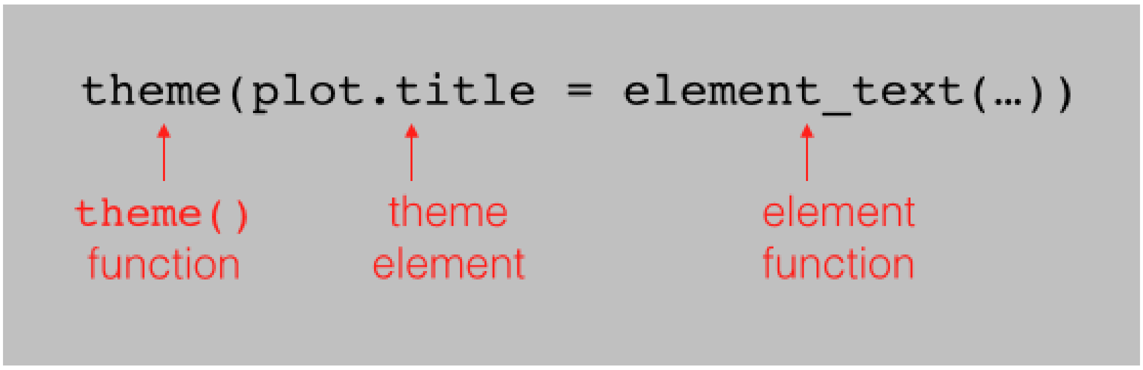
That’s the way that the theme() function works. You call it and specify the theme element of the plot that you want to modify.

Now look closer, inside each call of the theme() function. As I just noted, the first argument of the theme() function is the “theme element” that we want to modify.

But we’re specifying *how* we want to modify those theme elements using element_text(). Obviously, the axis title and plot titles are both “text.” When we modify “text” elements, we use the element_text() function to format them.

element_text() is an “element function” and it is used to format text elements.

That's how the ggplot2 theme system works: you call the theme() function; inside the theme function you specify the *theme element* you want to modify, and then you use an *element function* of the form "element_XYZ()." Inside the element function there are sets of parameters that control how we want to format the theme element.



```
theme(plot.title = element_text(…))
```

In this case, the plot and axis titles are "text" elements, so we call element_text().

Inside of element_text(), we're using several parameters to specify how we want to modify our titles: family, color, face, size (and for plot.title, we're additionally using hjust).

Here's a breakdown of the "text" parameters we're using within element_text().

[table width="500px"]
parameter, what it modifies
size=, font size
face=, bold/italic
color=, font color
hjust=, horizontal position
angle=, angle
[/table]

A special note about "inheritance"

Notice that the naming-conventions of the theme elements are *highly systematic*. Both plot.title and axis.title are "titles."

Like almost everything in ggplot2, there's a deep structure (Hadley Wickham did a fantastic job designing ggplot2). Once you understand the structure that's built into the system, things get easier.

You'll note that in the above code, we specified that we wanted to modify axis.title. That then filtered down to both the x and y axis titles.

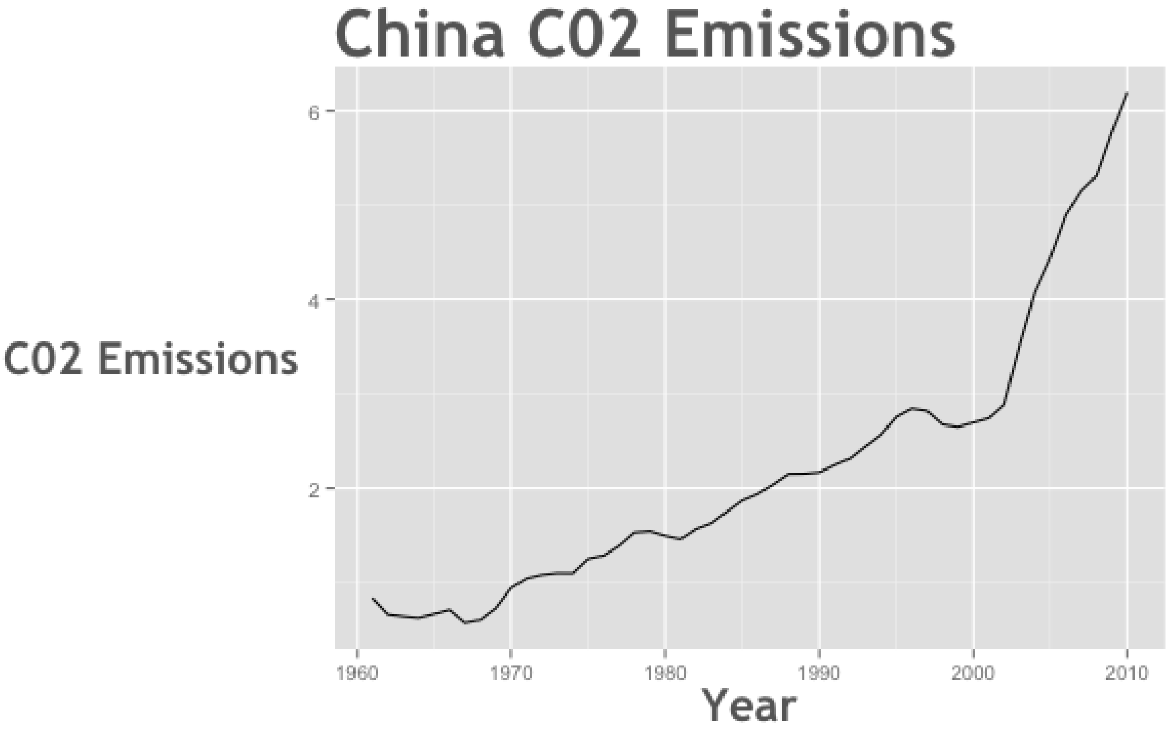
This is a case of "inheritance."

axis.title.x and axis.title.y are two separate theme elements. They inherit properties from axis.title.

That said, we can modify axis.title.x and axis.title.y individually (either to set a property for individual axes, or to override an inherited property).

Here’s a quick example. We’ll modify the *angle* of our y-axis title by calling theme() to format axis.title.y.

```
ggplot(data=df.china_co2, aes(x=year, y=co2_emission_per_cap_qt)) +
  geom_line() +
  ggtitle("China C02 Emissions") +
  labs(x="Year",y="C02 Emissions") +
  theme(plot.title = element_text(family = "Trebuchet MS", color="#666666",
face="bold", size=32, hjust=0)) +
  theme(axis.title = element_text(family = "Trebuchet MS", color="#666666",
face="bold", size=22)) +
  theme(axis.title.y = element_text(angle=0))
```



Here in this final example, we’ve set the “angle=” parameter for the axis.title.y element to “0.” This gives us a horizontal y-axis title.

Recap: How to format text elements (like titles)

So let’s take a quick step back.

Plots have “theme elements” which are just things that we want to format. We call the theme() function and specify the theme element that we want to format. We specify *how* we want to format that theme element by calling an “element function” and setting specific parameters inside of the element function.

To format “text elements” specifically, you:

1. Call theme()
2. Inside your theme() call, specify the plot element you want to format as your first argument

3. Then set the appearance of that element by calling element_text()
4. Inside of element_text(), set individual parameters to set the appearance of the element you want to format

Final note on workflow and the reason to learn this way

Ok, you might be thinking that this is a little complicated. You might even be tempted to go back to Excel and just do your plots there.

Don't. Really, the ggplot system has a little bit of a learning curve, but once you overcome that learning curve and learn things like the theme system, you'll be much more capable of producing great plots.

Moreover, the ggplot2 theme system can dramatically speed up your workflow. In a future article, I'm going to explain how to leverage the ggplot2 theme system to rapidly format your charts and save you time.

If you'd like to know when that walkthrough is published, then sign up for the email list now. I'll send all of my new material to you as an email subscriber, direct to your inbox, so you'll never miss a thing.