

The Complete Guide: Change Font Size in Base R Plots

Authored by
Mohammed looti

November 2, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *The Complete Guide: Change Font Size in Base R Plots*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=8392>

Creating effective data visualizations is crucial for communication, and often, the default settings in statistical software do not meet presentation standards. When working with [Base R Plots](#), adjusting visual elements like font size is essential for improving readability, especially when preparing charts for reports or academic papers.

This comprehensive guide details the exact syntax required to manipulate the font size of titles, subtitles, axis labels, and axis tick marks using the built-in graphical parameters within the [plot\(\) function](#). Mastering these simple arguments allows for precise control over your visualization aesthetics.

Understanding the Core Font Scaling Parameters

In Base R, font size adjustments are managed using the [graphical parameters](#), specifically those prefixed with `cex` (Character Expansion factor). These parameters accept a numeric value that scales the text size relative to the default size (where 1 represents the default). For instance, setting a parameter to 2 will double the font size.

The four primary `cex` arguments relevant to plot text elements are:

`cex.main`: Controls the font size of the main plot title.

`cex.sub`: Controls the font size of the plot subtitle.

`cex.lab`: Controls the font size of the X and Y axis labels (the descriptive text).

`cex.axis`: Controls the font size of the axis annotation (the numbers/tick mark labels).

You can apply all these parameters simultaneously within a single `plot()` call, as shown in the general syntax below:

```
plot(df$x, df$y, main='Title', sub='Subtitle',  
cex.main=2, #change font size of title  
cex.sub=2, #change font size of subtitle  
cex.lab=2, #change font size of axis labels  
cex.axis=2) #change font size of axis text
```

Setting Up the Demonstration Data

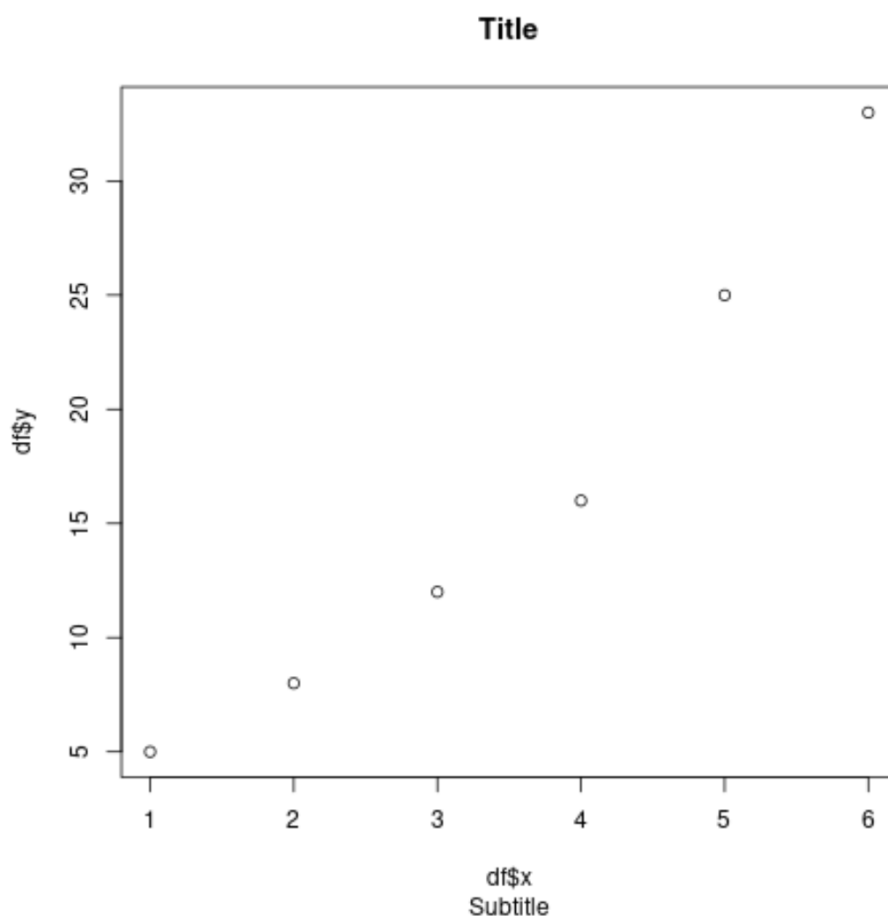
To illustrate the effect of these parameters, we will first create a simple [data frame](#) and generate a default [scatterplot](#). This baseline plot uses the standard R font sizes (i.e., all `cex` values are implicitly set to 1).

The following code defines our sample data, consisting of six (x, y) coordinates. This will be the foundational data structure used in all subsequent examples:

```
#create data frame
df <- data.frame(x=c(1, 2, 3, 4, 5, 6),
y=c(5, 8, 12, 16, 25, 33))

#create scatterplot with all default font sizes
plot(df$x, df$y, main='Title', sub='Subtitle')
```

The resulting visualization, shown below, serves as our benchmark for comparison against the modified plots:



Example 1: Adjusting the Main Plot Title (`cex.main`)

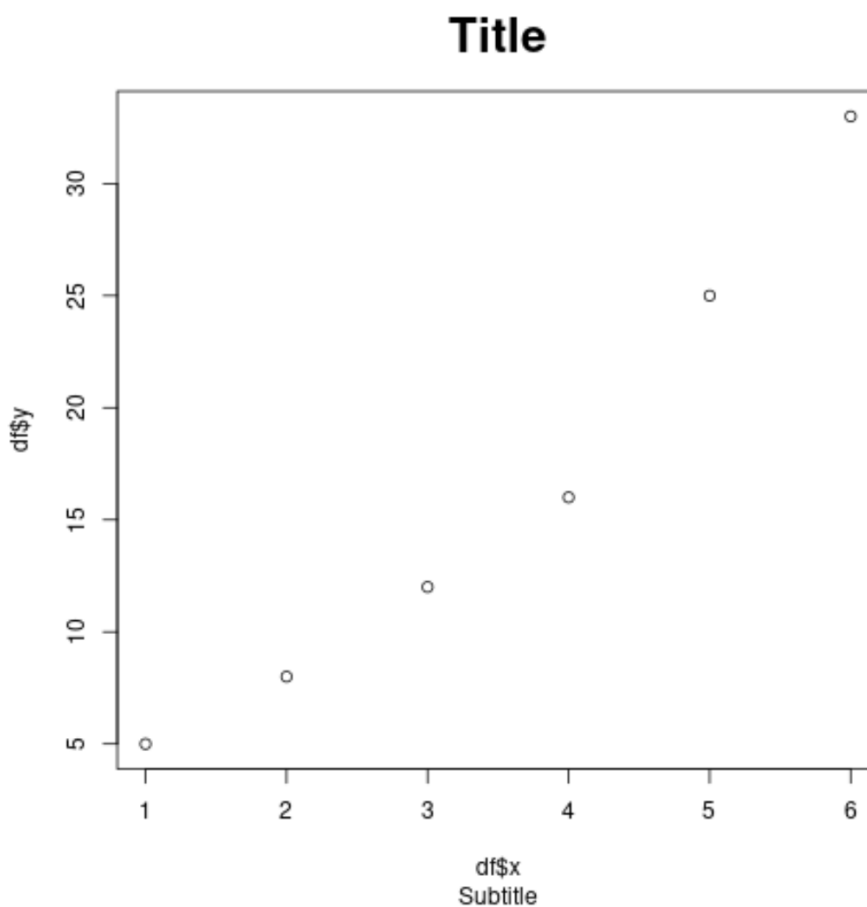
The main title of a plot is often the most important textual element, summarizing the entire visualization. If the title text is too small, it can be easily overlooked. We use the `cex.main` parameter to control its size.

In this example, we set `cex.main=2`, doubling the size of the title relative to the default setting. Note that other font elements (subtitle, axes) remain at their default size, highlighting the selective

application of this parameter.

```
#create scatterplot with increased font size of title  
plot(df$x, df$y, main='Title', sub='Subtitle',  
cex.main=2)
```

Observe how the title, "Title," is now significantly larger and more prominent, drawing immediate attention:



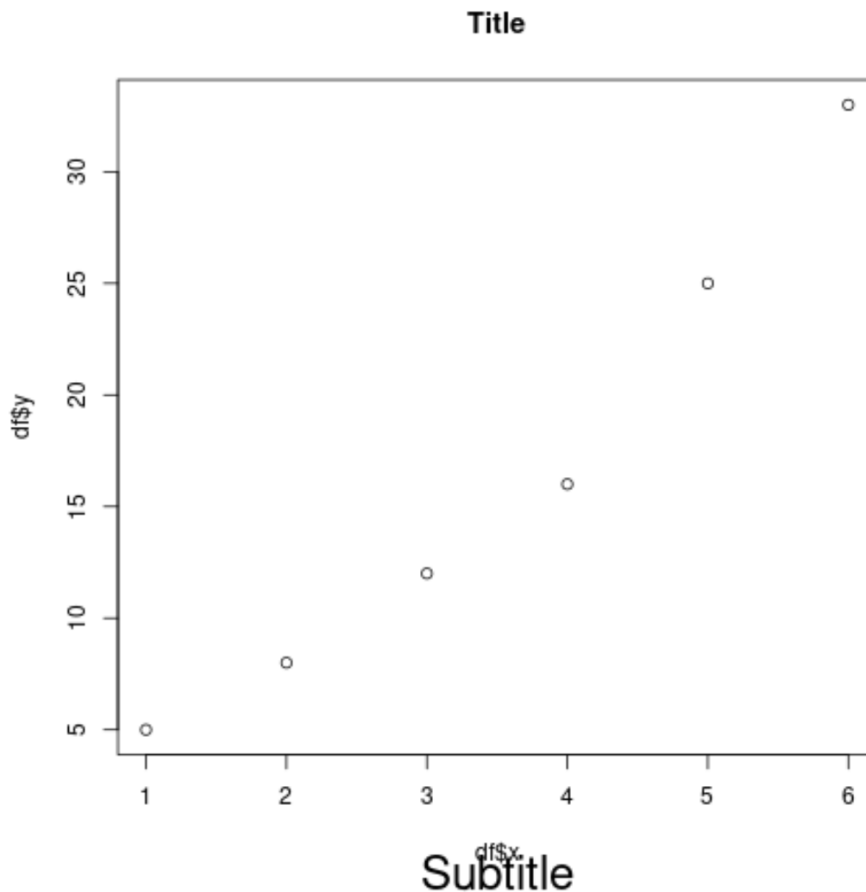
Example 2: Controlling the Subtitle Font Size (`cex.sub`)

Subtitles are useful for providing secondary information, data sources, or specific experimental details. While important, they should typically be smaller than the main title to maintain visual hierarchy. The `cex.sub` argument manages this element.

By setting `cex.sub=2`, we increase the subtitle size dramatically for demonstration purposes. Using this parameter allows the user to balance the prominence of the subtitle against the main title and the rest of the plot elements.

```
#create scatterplot with increased font size of subtitle  
plot(df$x, df$y, main='Title', sub='Subtitle',  
cex.sub=2)
```

The resulting plot clearly shows the enlarged subtitle, "Subtitle," confirming the effect of the `cex.sub` parameter:



Example 3: Scaling Axis Labels (`cex.lab`)

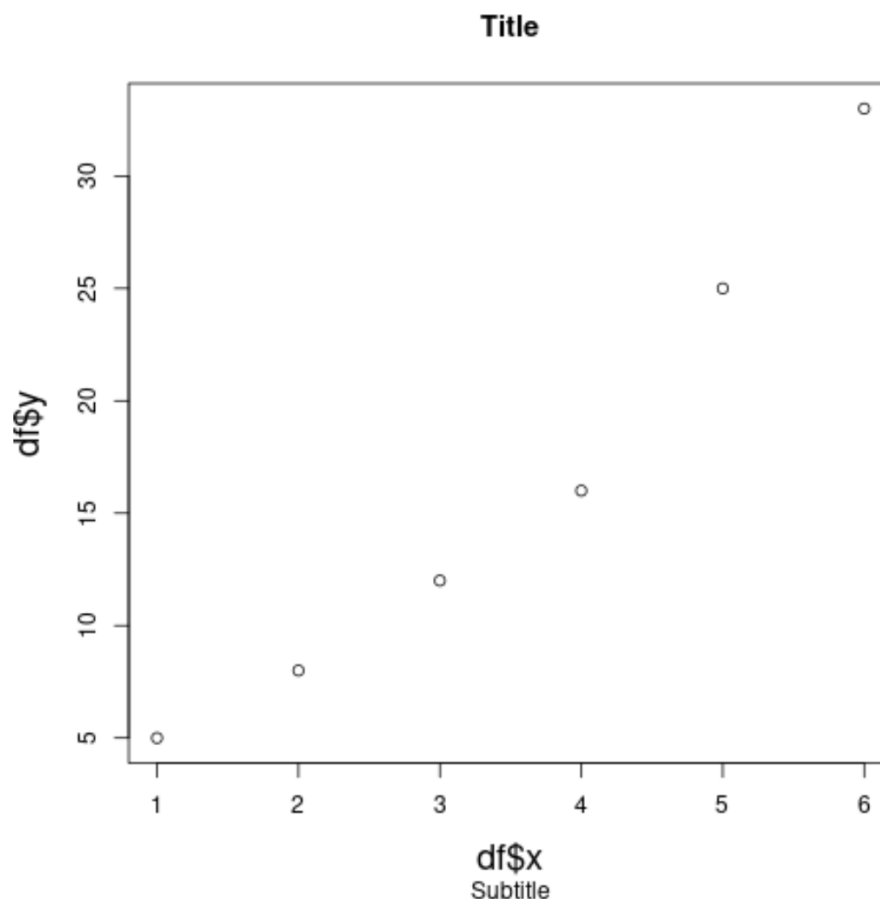
The [axis labels](#) (which describe the data shown on the axes, such as "x" and "y" in our example) are vital for interpreting the plot dimensions. If these labels are too small, the audience may struggle to understand what is being measured, rendering the plot ineffective.

The `cex.lab` parameter controls the font size of these descriptive labels. We again use a scale factor of 2 to ensure high visibility in our demonstration, illustrating how to prioritize the description of the variables.

```
#create scatterplot with increased font size of axis labels
```

```
plot(df$x, df$y, main='Title', sub='Subtitle',
     cex.lab=2)
```

Notice the increased size of the 'x' and 'y' axis descriptions, making the variables easier to identify and enhancing overall plot clarity:



Example 4: Changing Axis Tick Mark Annotations (`cex.axis`)

The axis text refers to the actual numerical values or categories displayed alongside the tick marks (also known as axis annotations). Readability of these numbers is paramount for accurate data reading, especially in complex plots or when dealing with small increments.

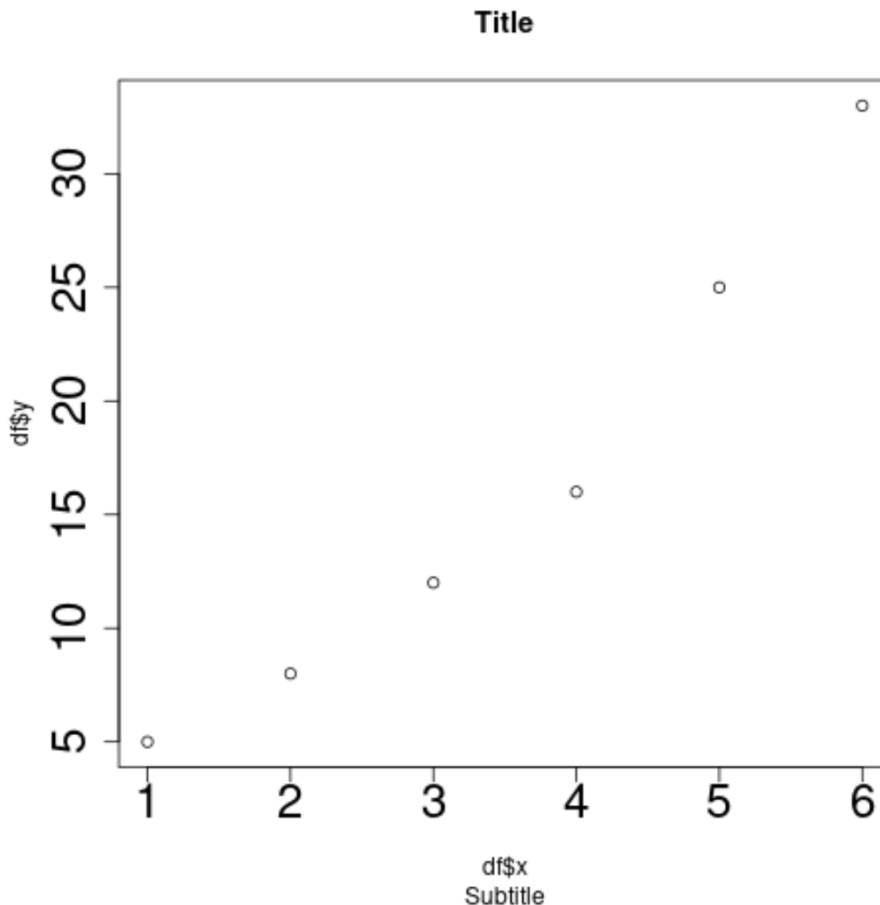
To control the size of these numerical annotations, we utilize the `cex.axis` parameter. Setting this value higher ensures that the quantitative scale of the [Base R plot](#) remains legible, preventing users from having to strain to read the specific values.

```
#create scatterplot with increased font size of axis text
```

```
plot(df$x, df$y, main='Title', sub='Subtitle',
```

`cex.axis=2)`

The resulting image displays significantly larger numbers along both the horizontal and vertical axes, improving the precision with which the audience can interpret the data points:



Summary and Further Customization

By using the specific `cex` graphical parameters--`cex.main`, `cex.sub`, `cex.lab`, and `cex.axis`--users can achieve precise control over the font sizes of all key textual elements in their R visualizations. Remember that these values are relative scaling factors; a value of 1.5 increases the size by 50%, while a value of 0.8 decreases it.

For even broader control over all text elements simultaneously, you can use the general `cex` argument within the `plot()` function, which scales all text elements uniformly. However, for professional reporting and fine-tuning individual elements for optimal hierarchy, the specific arguments demonstrated above are the optimal approach.

We encourage you to experiment with different scaling factors to find the ideal balance for your

specific presentation needs, ensuring your plots are both informative and aesthetically pleasing across all display sizes.

Additional Resources for R Plotting

To further enhance your skills in R visualization and perform other common plotting operations, consult the following related tutorials and documentation:

Official R Documentation on Graphical Parameters (`par()`)

Tutorial on customizing colors and background elements in Base R charts.

Guide to adding legends and annotations to R plots for advanced data presentation.