

Learning to Adjust Legend Size in Base R Plots: A Step-by-Step Guide

Authored by
Mohammed loot

October 29, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learning to Adjust Legend Size in Base R Plots: A Step-by-Step Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=5690>

Introduction: Mastering Legends in Base R Plots

Creating high-quality data visualizations is essential for effective statistical communication. A precisely designed [legend](#) is the key component that allows viewers to interpret complex plots accurately. In [Base R](#), the default graphical system provides robust tools for generating diverse visualizations, including [scatter plots](#), histograms, and bar charts. The primary role of the legend is to serve as a map, linking visual characteristics--such as colors, symbols, or line styles--to specific data categories or groups, thereby ensuring clarity and trustworthiness in your results.

Although the [Base R](#) environment offers extensive control over most plot aesthetics, adjusting the size of the legend is a frequent requirement, especially when aiming for publication-ready graphics. Default legend settings may sometimes clash with the overall scale or complexity of the visualization, resulting in a display that is either too crowded or difficult to read. Fortunately, users of [Base R](#) have direct, programmatic methods to customize these essential dimensions.

This expert guide focuses on the most effective technique for altering the size of a legend in a standard [Base R](#) plot: leveraging the powerful argument, `cex`, within the [legend\(\)](#) function. We will meticulously detail how this parameter uniformly scales the entire legend structure, encompassing both its text labels and graphical symbols, providing practical, step-by-step examples across different visualization contexts.

The `cex` Argument: Comprehensive Legend Scaling

The primary mechanism for controlling the overall dimensions of a legend in [Base R](#) is the `cex` argument. This abbreviation stands for "character expansion factor," and it is one of the foundational [Graphical Parameters](#) used throughout R's base plotting system. When integrated into the [legend\(\)](#) function, `cex` applies a uniform scaling factor to every element within the legend box. This includes the textual descriptions, the size of the associated symbols (like points or lines), and the internal spacing between legend items.

By default, the `cex` parameter is set to a value of `1`. This baseline ensures that all components within the legend are rendered at their standard size relative to the rest of the plot. To enlarge the legend elements, you must assign a value greater than `1`; the scaling will be directly proportional to the specified value (e.g., `2` doubles the size). Conversely, if the plot requires a more discrete legend, specifying a value less than `1` (such as `0.7`) will cause the entire legend block to shrink. This linear control offers predictable outcomes, simplifying the process of aligning the legend size with the visualization's overall aesthetic and spatial constraints.

The following snippet demonstrates the standard syntax for incorporating the `cex` argument into the [legend\(\)](#) function:

```
legend('topright', legend=c('A', 'B'), col=1:2, pch=16, cex=1)
```

In this basic setup, `'topright'` defines the placement on the plot, `legend=c('A', 'B')` provides the descriptive labels, `col=1:2` assigns colors, and `pch=16` selects the solid circle character. The final inclusion, `cex=1`, explicitly sets the legend to its default dimension. Modifying this single numeric value is the most direct way to control the visual prominence of your legend.

Practical Application: Adjusting Size in a Scatter Plot

To demonstrate the practical effect of the `cex` argument, we will work through a typical scenario: generating a [scatter plot](#) that visualizes data points belonging to distinct categorical groups. Our first step is to establish a sample R [data frame](#), which will serve as the foundation for our visual baseline.

Our illustrative [data frame](#), named `df`, is structured with three key variables: `x`, `y`, and `group`. The `group` variable is essential for differentiating the data points, allowing us to assign unique colors during plotting. We use the `plot()` function to create the scatter plot, using `col=df$group` to map colors to categories and `pch=16` for solid plot markers. Finally, the `legend()` function is called to generate a key, placed in the 'topright' quadrant, explaining the color-to-group mapping using default sizing.

#create data frame

```
df <- data.frame(x=c(1, 2, 3, 4, 5, 6),  
y=c(4, 6, 7, 12, 6, 8),  
group=c(1, 1, 1, 2, 2, 2))
```

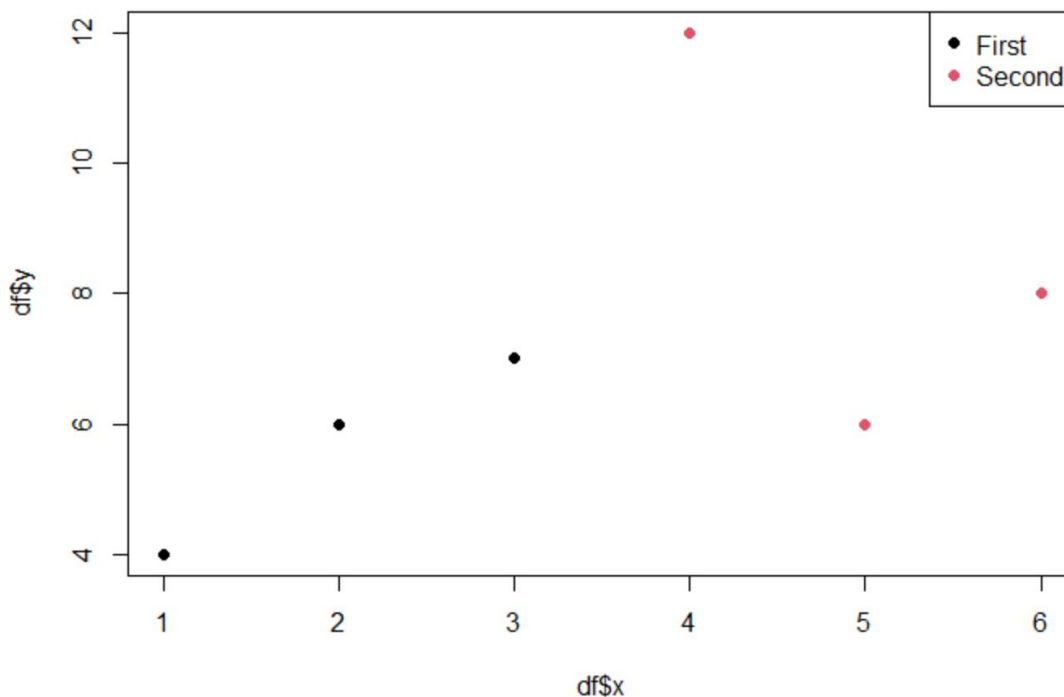
#create scatter plot

```
plot(df$x, df$y, col=df$group, pch=16)
```

#add legend in top right corner

```
legend('topright', legend=c('First', 'Second'),  
col=1:2, pch=16)
```

The resulting visualization, displayed below, features a clear [scatter plot](#) with two distinct data groups marked by red and black points. Critically, the accompanying legend is rendered using the system's default size. Take a moment to observe the initial proportion of the legend's text and symbols relative to the overall plot area, as this image establishes the necessary reference point for all subsequent size adjustments.



Enhancing Readability: Increasing Legend Size

If the default legend size proves insufficient--perhaps due to a high-resolution output, a large canvas, or a requirement for greater visual emphasis in a presentation--it is necessary to enlarge it. This modification is straightforwardly achieved by passing a value greater than 1 to the `cex` argument within the plotting call. For example, assigning `cex=2` will effectively double the size of all elements within the legend box compared to the standard default.

We will now implement this change using our existing R [data frame](#) and the scatter plot visualization. The only adjustment required is within the `legend()` call itself, where we explicitly define `cex` to be larger. This seemingly simple parameter tweak profoundly enhances the legibility of the legend, which is paramount for reports and visual media where immediate clarity is crucial for interpretation.

```
#create scatter plot
```

```
plot(df$x, df$y, col=df$group, pch=16)
```

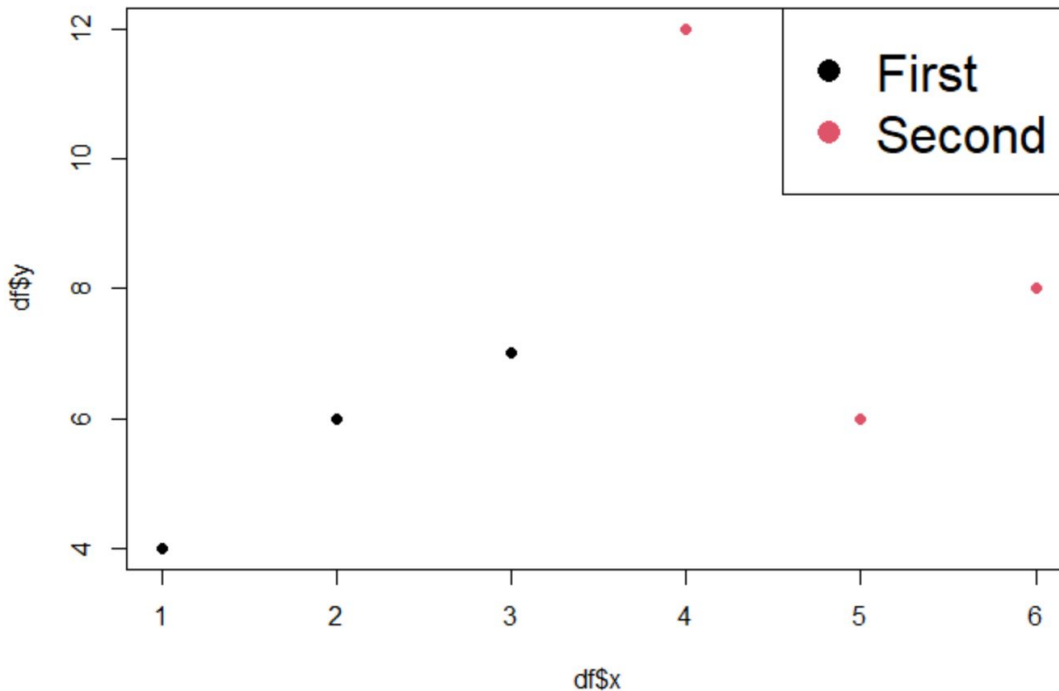
```
#add legend in top right corner with increased size
```

```
legend('topright', legend=c('First', 'Second'),
```

```
col=1:2, pch=16, cex=2)
```

As clearly demonstrated in the resulting image below, the legend has undergone significant

enlargement. Both the descriptive text labels ("First" and "Second") and their corresponding colored symbols have been scaled up uniformly. This outcome vividly illustrates the direct, proportional impact achieved by increasing the `cex` value, making the legend far more prominent and easier to read.



Optimizing Space: Decreasing Legend Size

Conversely, scenarios often arise where a smaller, more compact legend is necessary. For example, when dealing with plots containing dense data points, limited screen space, or if the legend contains verbose labels, a large legend can unfortunately obscure important graphical elements. In these instances, reducing the legend's dimensions is crucial for maximizing plot real estate and achieving a better overall visual balance.

To achieve this reduction in scale, we simply specify a numeric value less than `1` for the `cex` parameter. A practical choice like `cex=0.75` will render the legend at 75% of its default size. This technique allows you to effectively reclaim valuable space on the plotting canvas while ensuring that the necessary mapping information remains available in a concise format.

Let us modify our previous example to showcase this technique. We reuse the identical [data frame](#) and scatter plot setup, but this time, the `legend()` function will incorporate `cex=.75`. Observe how the legend gracefully scales down, placing greater focus on the visualized data while still maintaining its informational integrity.

```
#create scatter plot
```

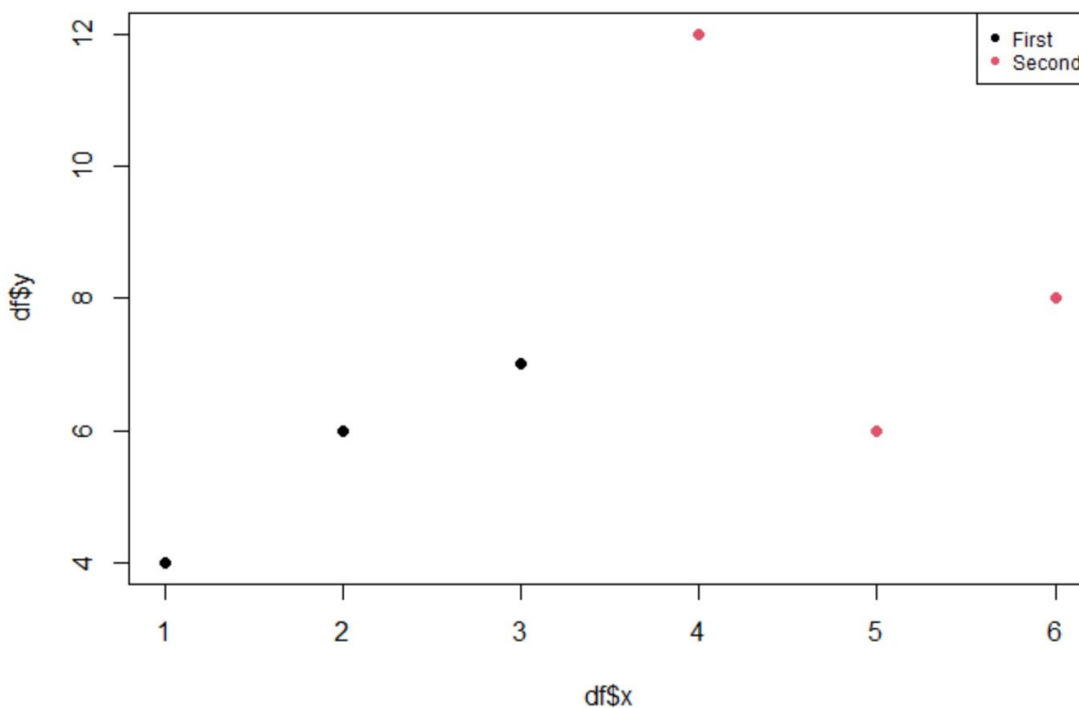
```
plot(df$x, df$y, col=df$group, pch=16)
```

```
#add legend in top right corner with decreased size
```

```
legend('topright', legend=c('First', 'Second'),
```

```
col=1:2, pch=16, cex=.75)
```

The plot displayed below clearly demonstrates the outcome of using a smaller `cex` value. The legend's textual elements and symbols are noticeably more compact. This specific adjustment is highly valuable when integrating plots into academic papers or reports where space constraints are severe, or when the primary focus needs to remain overwhelmingly on the data pattern itself, rather than the explanatory key.



Granular Control: Scaling Symbols with `pt.cex`

While `cex` offers global scaling for the entire legend, there are specialized requirements where only the symbols (points, lines, or icons) within the legend need size adjustment, without affecting the text labels or the overall spatial layout. For achieving this highly granular control, [Base R](#) provides the `pt.cex` argument, which stands for "point character expansion factor," integrated directly into the `legend()` function.

The `pt.cex` argument functions identically to `cex` in accepting numerical values to scale the

symbols. A value of `1` maintains the default size, while values above `1` increase symbol size and values below `1` decrease it. The crucial difference is its scope: `pt.cex` operates entirely independently of the legend's text size. This separation allows for intricate fine-tuning, ensuring that symbols remain visually distinct and discernible even if the corresponding legend text is minimized, or vice versa.

Let's observe this unique functionality. We will maintain the overall legend size at its default (or any chosen `cex` value) and specifically magnify the size of the legend points using `pt.cex=2`. This setting will double the size of the colored points in the legend, but the textual labels ("First" and "Second") will remain unchanged, adhering to the standard text size.

#create scatter plot

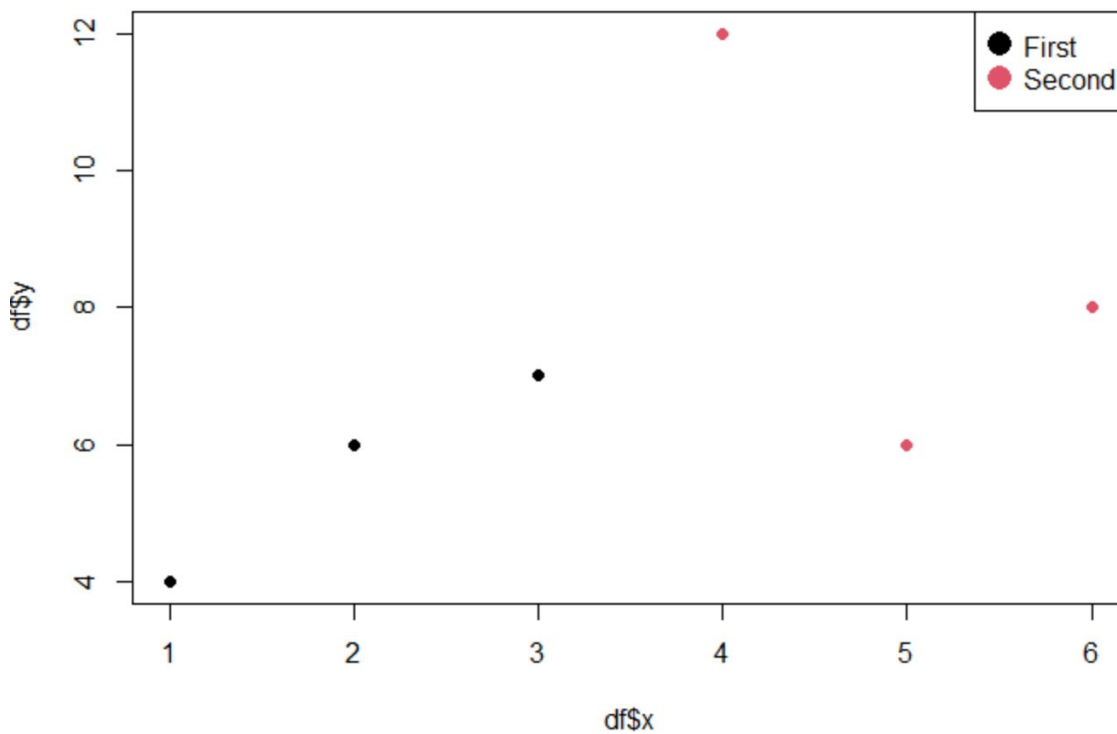
```
plot(df$x, df$y, col=df$group, pch=16)
```

```
#add legend in top right corner with increased point size
```

```
legend('topright', legend=c('First', 'Second'),
```

```
col=1:2, pch=16, pt.cex=2)
```

Upon reviewing the plot below, the difference is immediate and striking. The overall structure, including the text size, adheres to the default proportions (as `cex` was implicitly `1`). However, the red and black points acting as visual identifiers within the legend are now distinctly larger due to the applied `pt.cex` argument. This ability to independently control graphical elements enables precise management of the legend's visual hierarchy and effectiveness.



Conclusion and Advanced Customization Techniques

Mastering legend size management is a fundamental skill for generating clear, professional statistical visualizations in R. The `cex` argument provides an efficient, global scaling mechanism for all legend components, facilitating rapid adjustments to its overall visibility and footprint. For situations demanding finer control, the `pt.cex` argument offers the specialized ability to scale only the symbols (points, lines, etc.), ensuring maximum clarity regardless of the text size chosen.

Beyond simple sizing, the `legend()` function includes an extensive range of customization options. Users can precisely define placement (e.g., using keywords like `'topleft'` or exact coordinates), incorporate a title, modify the background fill (using the `bg` parameter), specify border visibility (`bty`), and adjust the spacing between symbols and text (`x.intersp` and `y.intersp`). Exploring these [Graphical Parameters](#) will significantly enhance your ability to create legends that perfectly integrate with your visualizations.

We strongly recommend experimenting with these arguments and consulting the official [R documentation](#) for the `legend()` function. Achieving proficiency in legend customization is a crucial step toward elevating the quality and interpretability of your statistical graphics output.

Additional Resources for R Plot Customization

To further optimize your skills in crafting compelling data visualizations using R's base graphics

system, we encourage you to explore these specialized tutorials and guides:

[How to Change Font Size in R Plots](#)

[How to Add a Title to a Plot in R](#)

[How to Change Axis Labels in R](#)

[Understanding Color Palettes in R for Data Visualization](#)

[Customizing Plot Margins in Base R Graphics](#)