How to Create a Horizontal Legend in Base R (2 Methods)

Authored by

stats writer

November 19, 2025

Creating professional-quality graphics in Base R requires precise control over all visual elements, particularly the legend, which is crucial for interpreting complex data visualization. While R typically defaults to a vertical legend format, this can often be inefficient, consuming vital vertical space, especially in layouts designed for compact display. This comprehensive guide details two expert-level methods for generating a robust, space-saving horizontal legend using the primary legend() function. The first method leverages the simple **horiz** argument to place all items on a single row, providing maximum compression. The second method, utilizing the **ncol** argument, offers flexibility by arranging items into multiple compact columns, ensuring readability even when the number of categories is large. Both approaches, when paired with careful placement parameters like **inset** and **xpd**, allow the user to optimize the graphical output for clarity and aesthetic appeal, ultimately enhancing the communication of analytical results.

To achieve effective placement and orientation for your explanatory key in Base R graphics, focus on these two primary methods within the **legend()** function. Mastering these parameters ensures your visual data remains the focal point while the categorical labels are clearly presented without compromising the primary plotting area:

Method 1: Use the horiz argument

```
legend('bottom', fill=fill_cols, legend=c('A', 'B', 'C', 'D', 'E', 'F'),
horiz=TRUE, inset=c(0, -.1), xpd=TRUE)
```

This particular example creates a truly horizontal legend positioned below the plot, guaranteeing that each item in the legend is aligned on the single row. This technique is highly recommended for situations where you have a small to moderate number of categories and prioritize maximum horizontal compression.

Method 2: Use the ncol argument

```
legend('bottom', fill=fill_cols, legend=c('A', 'B', 'C', 'D', 'E', 'F'),
ncol=3, inset=c(0, -.15), xpd=TRUE)
```

This alternative approach constructs a horizontally distributed legend below the plot by arranging the items into three columns. This provides a balance between horizontal placement and space management, ensuring the legend fits within the plot width even with a larger number of entries that might otherwise stretch off-screen in a single row.

The **inset(x, y)** argument controls the precise location of the legend relative to the margin. By utilizing negative values for the y-component, we can strategically push the legend down outside of the standard plot bounding box, placing it cleanly in the outer margin space.

The **xpd=TRUE** argument is essential when using negative insets, as it permits plotting outside the default boundaries. This crucial graphical parameters (par) setting allows the legend to be drawn into the outer plotting device area and still be visible, preventing clipping and ensuring the full legend is displayed.

The following examples showcase the practical implementation of each method, demonstrating how these parameters work together to produce publication-ready graphics.

Example 1: Use horiz Argument to Create a Horizontal Legend in Base R

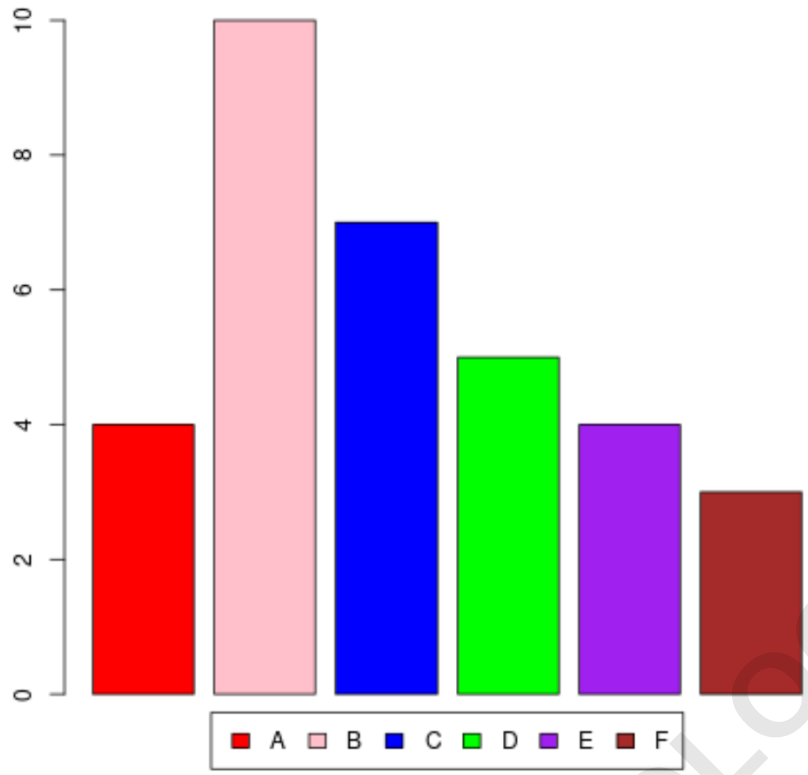
The code below illustrates the use of the **horiz=TRUE** argument within the **legend()** function. This method is the definitive way to achieve a single-row layout, ensuring that all legend entries are forced onto one line. We first establish a simple vector of data and corresponding fill colors, then generate a bar plot. The subsequent **legend()** call specifies positioning at the **'bottom'**, enabling horizontal display, and utilizing the critical **inset** and **xpd** parameters for precise placement outside the chart area.

```
#create vector of values
data <- c(4, 10, 7, 5, 4, 3)

#specify fill colors to use
fill_cols <- c('red', 'pink', 'blue', 'green', 'purple', 'brown')

#create bar plot to visualize values in vector
barplot(data, col=fill_cols)

#add legend to bottom of plot
legend('bottom', fill=fill_cols, legend=c('A', 'B', 'C', 'D', 'E', 'F'),
horiz=TRUE, inset=c(0, -.1), xpd=TRUE)
```



Notice that a perfectly horizontal legend has been generated and is successfully placed in the margin area at the bottom of the plot. This result confirms that **horiz=TRUE** is the ideal method for creating a maximally compressed legend when the number of items is manageable within the plot's width.

For greater control over the visual separation between the chart and the key, you can manipulate the values in the **inset** argument to adjust the exact vertical location of the legend below the x-axis. This flexibility is crucial for maintaining visual hierarchy in complex graphics.

For example, we can make the y-value for the **inset** argument more negative--changing **-.1** to **-.2**--to push the legend further down into the outer margin, creating more vertical buffer space:

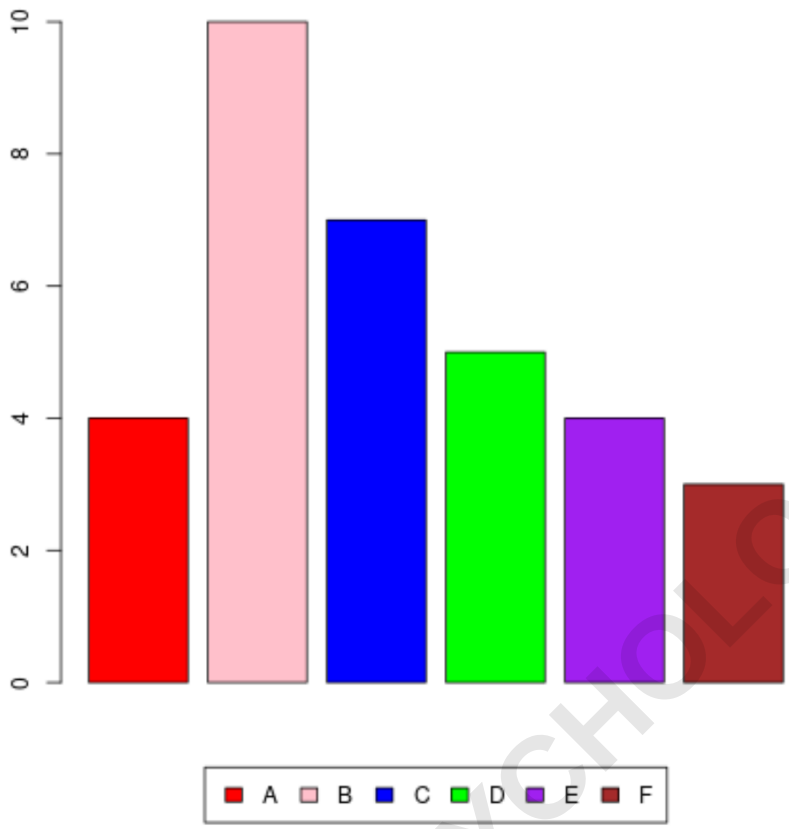
```
#create vector of values
data <- c(4, 10, 7, 5, 4, 3)

#specify fill colors to use
fill_cols <- c('red', 'pink', 'blue', 'green', 'purple', 'brown')

#create bar plot to visualize values in vector
barplot(data, col=fill_cols)

#add legend to bottom of plot
```

```
legend('bottom', fill=fill_cols, legend=c('A', 'B', 'C', 'D', 'E', 'F'),
horiz=TRUE, inset=c(0, -.2), xpd=TRUE)
```



Observe how the horizontal legend has been pushed significantly lower down below the plot, demonstrating the precise control afforded by carefully tuning the **inset** parameter alongside the **xpd=TRUE** setting.

Example 2: Use ncol Argument to Create a Horizontal Legend in Base R

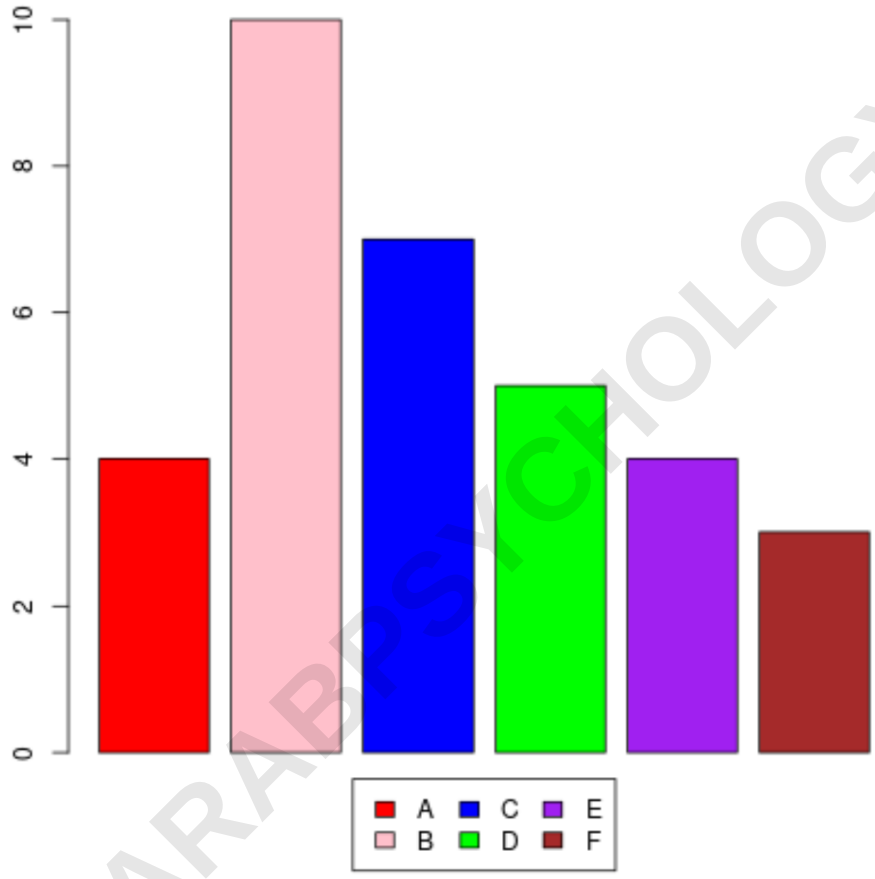
When dealing with a larger number of categories, using a single row (**horiz=TRUE**) can stretch the legend beyond the visual confines of the plot. The solution is to use the **ncol** argument, which allows us to specify the number of columns, thereby creating a multi-row, column-based structure that still reads horizontally. The following code uses **ncol=3** to arrange the six legend items into a two-row by three-column layout, maintaining a compact horizontal footprint while reducing the overall length of the element:

```
#create vector of values
data <- c(4, 10, 7, 5, 4, 3)
```

```
#specify fill colors to use
fill_cols <- c('red', 'pink', 'blue', 'green', 'purple', 'brown')

#create bar plot to visualize values in vector
barplot(data, col=fill_cols)

#add legend to bottom of plot
legend('bottom', fill=fill_cols, legend=c('A', 'B', 'C', 'D', 'E', 'F'),
ncol=3, inset=c(0, -.15), xpd=TRUE)
```



Notice that a horizontal legend with three columns has been successfully created and is placed neatly at the bottom of the plot. This demonstrates how the **ncol** approach achieves a horizontally efficient layout suitable for medium to large lists of categories.

Developers are encouraged to experiment with the value for the **ncol** argument to determine the optimal column arrangement for their specific data visualization needs, adjusting the number of columns to maximize the visual appeal and readability within the constraints of the available margins.

Summary of Horizontal Legend Creation

In summary, both the **horiz=TRUE** and the **ncol** arguments provide robust mechanisms for converting the default vertical legend into an organized, horizontally focused display in Base R. The choice between the two methods hinges on the number of categories and the amount of horizontal space available in the target environment. For concise lists, **horiz=TRUE** offers maximal compression. For longer lists, leveraging **ncol** ensures the legend remains compact and centered. In all cases, the use of **inset** and **xpd=TRUE** is paramount for placing the element outside the main plot, thereby preserving the clarity and integrity of the visualized data.

By mastering these core graphical parameters (par), users can significantly elevate the quality and professional appearance of their R plots, ensuring that the legend complements, rather than detracts from, the visual presentation of their analytical findings.