How to Change Axis Labels of Boxplot in R (With Examples)

Authored by

stats writer

November 22, 2025

Understanding Boxplots and Axis Customization in R

The Boxplot, or box-and-whisker plot, is a fundamental statistical visualization tool used widely in R. It provides a concise summary of the distribution of a dataset, highlighting key statistical measures such as the median, quartiles, and potential outliers. While R excels at generating these plots quickly, creating publication-ready or presentation-quality graphics often requires careful customization, especially concerning axis labels and titles. The default labels generated by R often rely on variable names, which may not be descriptive enough for external audiences.

Fortunately, R offers robust mechanisms for controlling graphical parameters. When working with base graphics, the standard method for customizing axis labels involves specifying the **xlab** and **ylab** arguments within the plotting function. The **xlab** argument controls the label applied to the horizontal (x) axis, while **ylab** manages the label for the vertical (y) axis. Furthermore, the **main** argument is universally available across many plotting functions in Base R to define a comprehensive title for the entire visualization.

Mastering these customizations is essential for effective data storytelling. Whether you are using the traditional Base R plotting system or leveraging the powerful, grammar-of-graphics approach provided by the **ggplot2** package, the ability to clearly define what each axis represents transforms raw graphical output into meaningful insight. This guide details the step-by-step processes for customizing boxplot axis labels using both of these predominant R visualization frameworks, ensuring your visualizations are both accurate and accessible. We will focus specifically on how to replace default labels with custom, descriptive text.

Overview of Label Customization Methods

When aiming to modify the labels displayed beneath the individual boxplots (the x-axis labels), there are fundamental differences in approach depending on which graphical system you choose. Both Base R and ggplot2 provide sophisticated ways to handle categorical data labeling, but the functions utilized vary significantly. The key distinction lies in how the data structure interacts with the plotting parameters. You can use one of the following standard methods to change the x-axis labels on a boxplot in R:

Method 1: Change Axis Labels of Boxplot in Base R

boxplot(df, names=c('Label 1', 'Label 2', 'Label 3'))

Method 2: Change Axis Labels of Boxplot in ggplot2

levels(df_long$variable) <- c('Label 1', 'Label 2', 'Label 3')

```
ggplot(df_long, aes(variable, value)) +
geom_boxplot()
```

Data Preparation: Setting up the Example Data Frame

To effectively demonstrate both customization methods, we will generate a sample data frame containing three distinct variables (A, B, and C). These variables represent different groups or conditions, which is the typical scenario requiring unique axis labeling. We utilize the **set.seed()** function to ensure that the random data generated remains consistent every time the code is executed, guaranteeing reproducibility of the examples shown below.

The data frame, `df`, is constructed such that each column contains 1,000 observations drawn from a normal distribution, with different specified means (5, 10, and 15, respectively). This difference in means will ensure that the resulting boxplots are visually distinct, making the effect of the label changes readily apparent. This initial setup in the wide format is suitable for Base R plotting, but will need conversion later for ggplot2. The following examples show how to use each method in practice with this data frame in R:

```
#make this example reproducible
set.seed(0)

#create data frame
df <- data.frame(A=rnorm(1000, mean=5),
B=rnorm(1000, mean=10),
C=rnorm(1000, mean=15))

#view head of data frame
head(df)

A B C
1 6.262954 9.713148 15.44435
2 4.673767 11.841107 15.01193
3 6.329799 9.843236 14.99072
4 6.272429 8.610197 14.69762
5 5.414641 8.526896 15.49236
6 3.460050 9.930481 14.39728
```

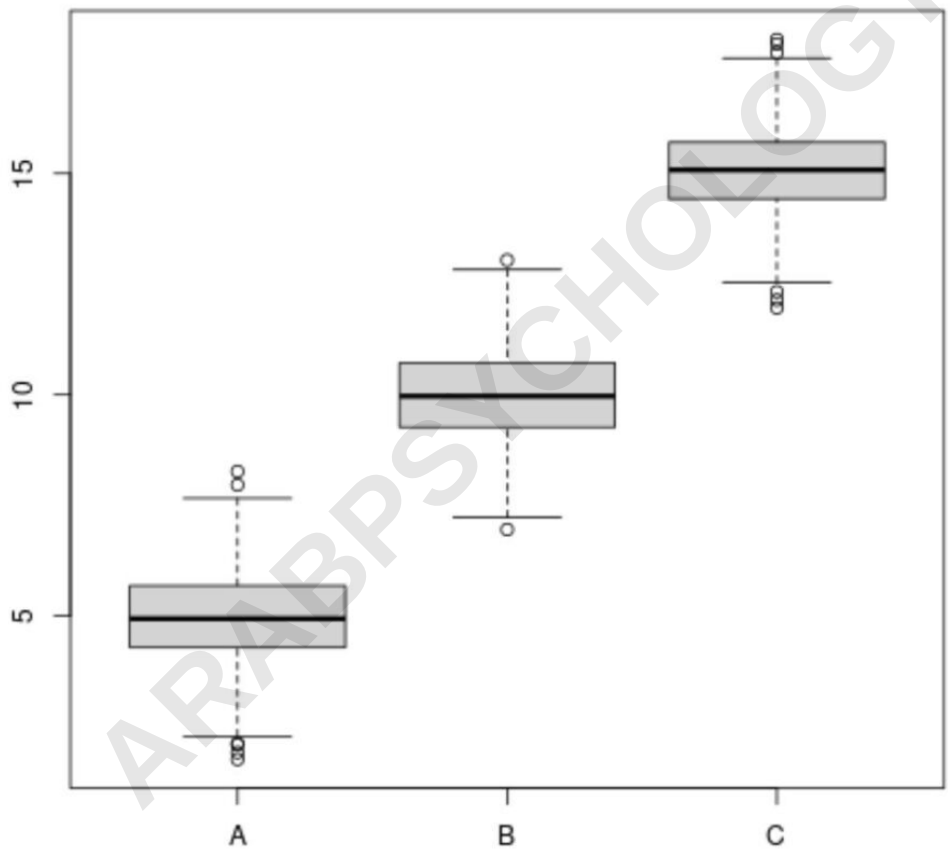
Example 1: Customizing Axis Labels in Base R

When utilizing the standard graphics capabilities provided by Base R, the **boxplot()** function is the

primary tool for visualization. By default, when supplied with a wide-format data frame (like `df`), the function automatically interprets each column as a separate group and uses the corresponding column names (A, B, C) as the labels along the x-axis. While convenient for initial exploration, these abbreviated labels are often insufficient for publication purposes, necessitating clearer descriptive text.

The following code snippet demonstrates the default behavior. If we use the **boxplot()** function to create boxplots in Base R, the column names of the data frame will be used as the x-axis labels by default:

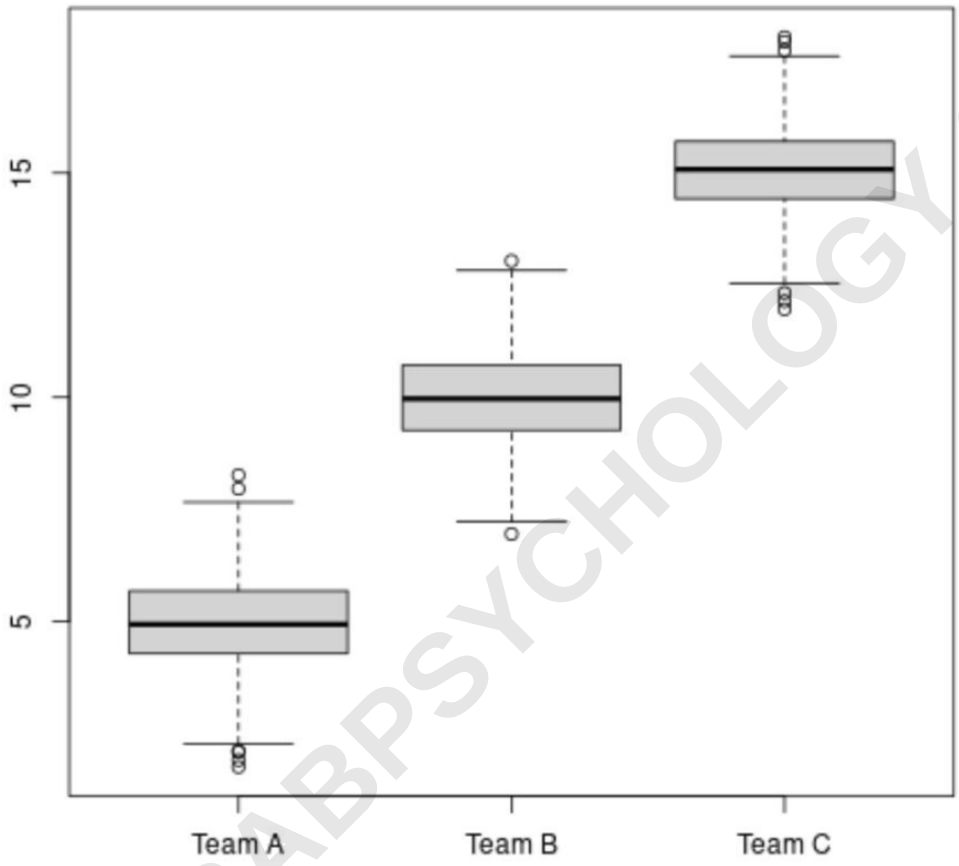
```
#create boxplots
boxplot(df)
```



To override the default column names, Base R provides the **names** argument within the **boxplot()** function. This argument accepts a character vector where each element corresponds sequentially to the boxplot groups generated from the data frame columns. By passing a vector of meaningful strings--such as 'Team A', 'Team B', and 'Team C'--we can instantly enhance the clarity of the visualization without altering the underlying data structure.

However, we can use the **names** argument to specify the x-axis labels to use. It is important that the number of elements in the supplied vector exactly matches the number of groups (columns) being plotted, otherwise, the function will return an error or unexpected results. This technique allows for rapid prototyping and effective labeling within Base R's graphical environment.

#create boxplots with specific x-axis names
boxplot(df, names=c('Team A', 'Team B', 'Team C'))



As clearly illustrated by the resulting plot, the labels we specified in the **names** argument ('Team A', 'Team B', 'Team C') successfully replace the default column headers along the horizontal axis. This confirms the efficacy of using the **names** parameter for fine-grained control over categorical labels in Base R visualizations.

Example 2: Changing Axis Labels of Boxplot in ggplot2

While Base R offers simplicity, the **ggplot2** package is generally preferred for creating complex, aesthetic, and layered visualizations. However, ggplot2 adheres strictly to the "tidy data" principle,

requiring data to be in a long format where observations are organized vertically. This structure necessitates a crucial preprocessing step when starting from a wide-format data frame like our `df`.

Before we can create boxplots in ggplot2, we must use the **melt()** function from the **reshape2** package to "melt" the data frame into a long format. Melting transforms the column headers into a single categorical column (`variable`) and the observations into a single value column (`value`). This reshaping step is mandatory because ggplot2 visualizes relationships between variables mapped to specific aesthetics.

library(reshape2)

#reshape data frame to long format
df_long <- melt(df)

#view head of long data frame
head(df_long)

variable value
1 A 6.262954
2 A 4.673767
3 A 6.329799
4 A 6.272429
5 A 5.414641
6 A 3.460050

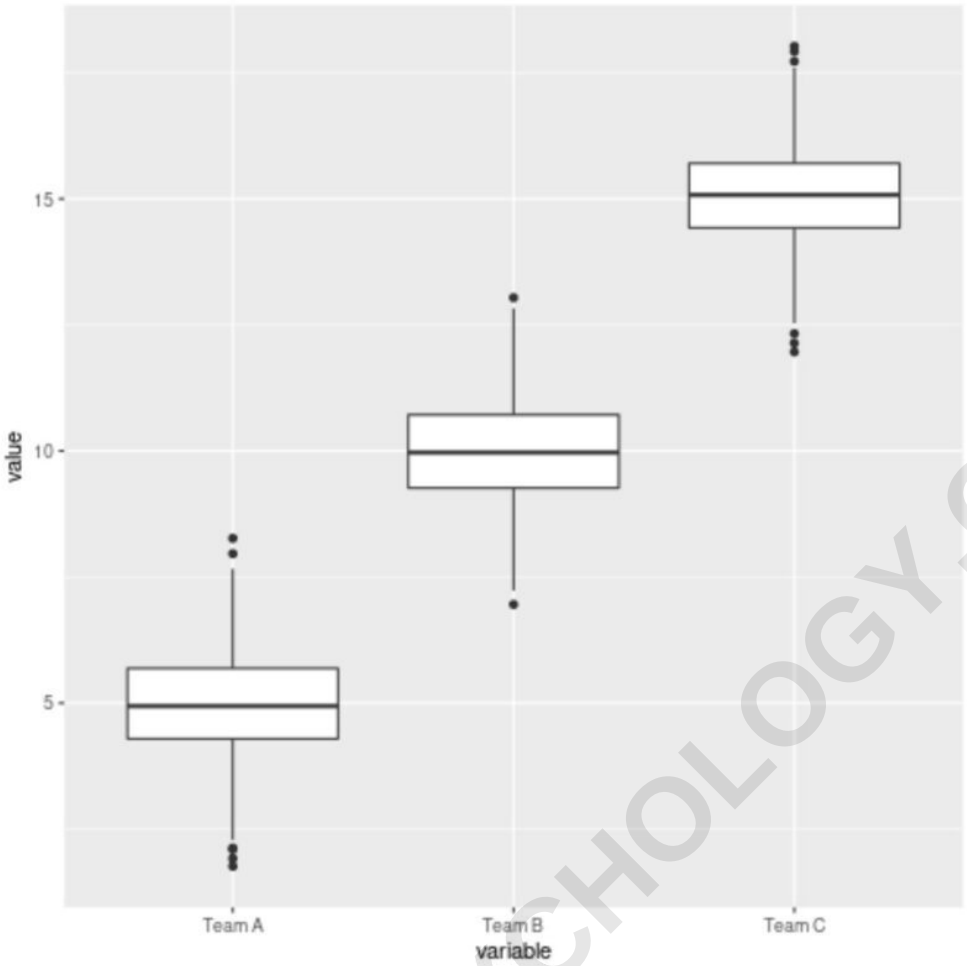
Once the data is in the long format (`df_long`), the 'variable' column, which contains 'A', 'B', and 'C', is treated by R as a factor. The labels displayed on the final boxplot correspond directly to these factor levels. To change the displayed labels without altering the underlying data categories, we use the **levels()** function to reassign the descriptive names to the existing levels.

We can then use the **levels()** function to specify the x-axis labels and the **geom_boxplot()** function to actually create the boxplot in ggplot2. This method of label customization is highly robust, ensuring that the labels are treated correctly throughout the plotting pipeline.

library(ggplot2)

#specify x-axis names to use
levels(df_long$variable) <- c('Team A', 'Team B', 'Team C')

#create box plot with specific x-axis labels
ggplot(df_long, aes(variable, value)) +
geom_boxplot()



The final output confirms that the labels we specified using the **levels** function--'Team A', 'Team B', and 'Team C'--are now used as the x-axis labels, replacing the original single-letter identifiers. This demonstrates the effective use of factor level manipulation in the ggplot2 environment for achieving highly customized and meaningful categorical labeling.