Plot Multiple Boxplots in One Chart in R

Authored by

Mohammed looti

November 9, 2025

Understanding the Box Plot: A Foundation for Data Visualization

A Box plot, often referred to as a box-and-whisker plot, is a powerful statistical tool used in data visualization to display the distribution of a numerical dataset. Its primary utility lies in providing a concise visual representation of the data's central tendency, spread, and symmetry, while also highlighting potential outliers. This chart condenses vast amounts of data into a recognizable format based on the five-number summary.

Understanding the components of the box plot is crucial for interpreting data distributions correctly. The core box shape represents the interquartile range (IQR), while the whiskers extend to show the variability outside this range. This structure makes box plots highly effective for comparing distributions across different groups or categories, which is the focus of this tutorial using the powerful statistical programming language, R.

The core components that define the structure of the box plot are derived from the five-number summary:

The **minimum value** (excluding outliers).
The **first quartile** (Q1), representing the 25th percentile.
The **median value** (Q2), marking the 50th percentile.
The **third quartile** (Q3), representing the 75th percentile.
The **maximum value** (excluding outliers).

This tutorial will guide you through the process of generating and customizing multiple boxplots within a single chart in R, utilizing both the built-in plotting functions of base R and the flexible capabilities of the popular extension package, ggplot2.

Plotting Boxplots Using Base R Graphics

The base graphics system in R provides a straightforward mechanism for creating standard statistical plots, including the box plot, using the simple `boxplot()` function. To demonstrate how to create both single and multiple comparative box plots, we will utilize the publicly available **airquality** dataset, which is conveniently built into R and contains daily air quality measurements in New York from May to September 1973. This dataset is ideal because it contains both continuous variables (like Ozone and Temp) and categorical variables (like Month).

Before generating any visualizations, it is good practice to inspect the structure and initial rows of the dataset to understand the data types and column names. We use the `head()` function to view the first six observations of the **airquality** dataset:

#view first 6 rows of "airquality" dataset

head(airquality)

Ozone Solar.R Wind Temp Month Day
#1 41 190 7.4 67 5 1
#2 36 118 8.0 72 5 2
#3 12 149 12.6 74 5 3
#4 18 313 11.5 62 5 4
#5 NA NA 14.3 56 5 5
#6 28 NA 14.9 66 5 6

The initial inspection confirms that we have several variables suitable for box plotting, including **Ozone** (mean ozone concentration), **Temp** (average temperature), and the categorical variable **Month** (coded 5 through 9 for May through September).

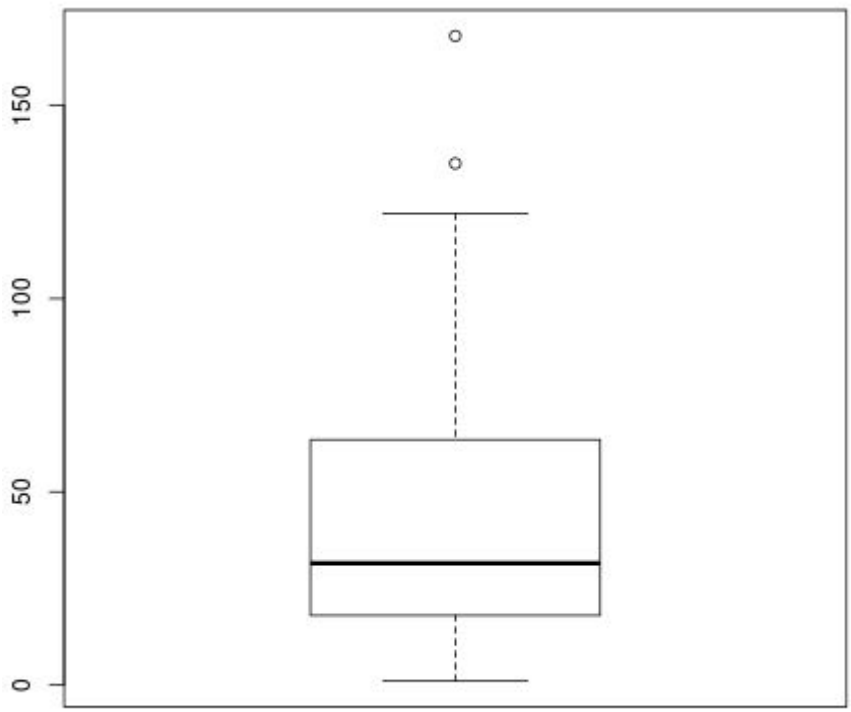
Creating Single-Variable Boxplots in Base R

The most fundamental application of the `boxplot()` function is to visualize the distribution of a single continuous variable. For instance, if we wish to examine the spread and central tendency of the Ozone levels across the entire dataset, we simply pass the column vector to the function. This provides an immediate summary of the data, highlighting the median, the quartiles, and any extreme values that might be considered outliers.

To create a single box plot for the variable "Ozone", we use the following straightforward syntax, specifying the column from the **airquality** data frame:

#create boxplot for the variable "Ozone"
boxplot(airquality$Ozone)

This command generates a visualization that effectively communicates the overall spread and symmetry of the ozone concentration data, allowing for quick identification of the median concentration and the range within which the central 50% of observations fall.



Generating Multiple Boxplots for Grouped Data in Base R

The true power of comparative box plotting emerges when we analyze how a continuous variable's distribution shifts across different categories. In R's base plotting system, this is achieved using the formula interface, which typically takes the form `Y ~ X`, where Y is the continuous variable whose distribution we are measuring, and X is the categorical variable defining the groups.

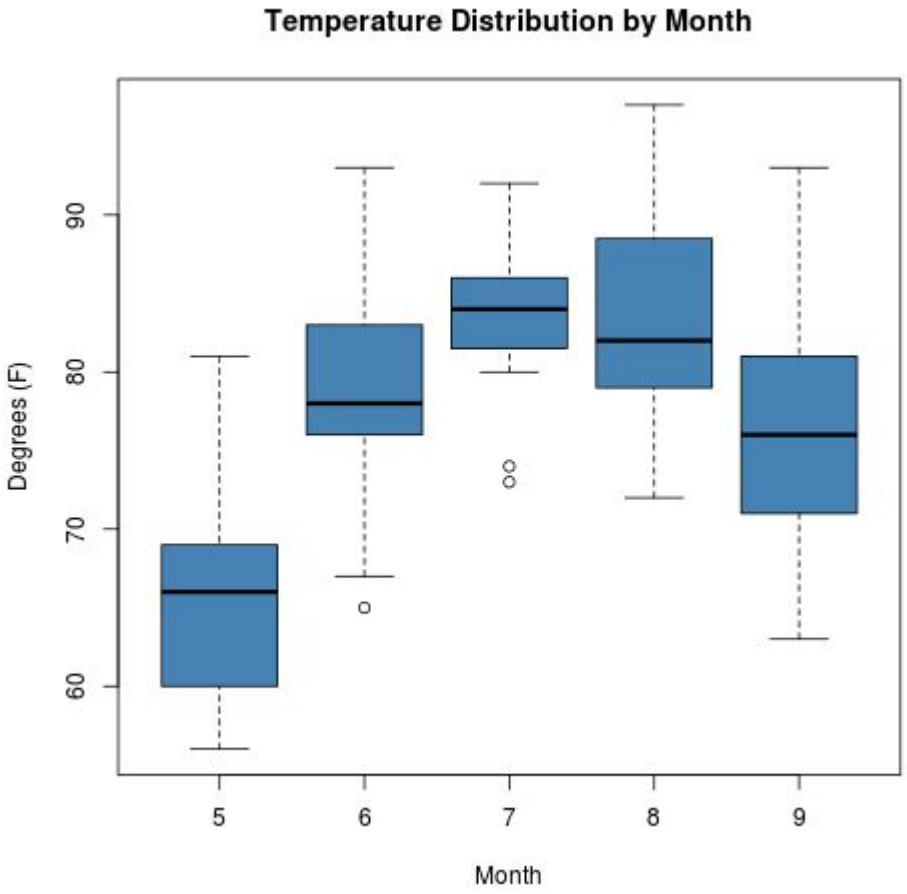
Suppose our goal is to compare the distribution of **Temperature** (Temp) across each **Month** (Month) in the dataset. This requires generating five separate box plots--one for each month from May (5) through September (9)--and displaying them side-by-side on the same chart for easy comparison. The formula `Temp~Month` tells R to partition the temperature data according to the month variable.

We can enhance the resulting chart by including standard graphical parameters such as a title (`main`), axis labels (`xlab` and `ylab`), and visual styling elements like color (`col`) and border thickness (`border`). Customization is key to creating publication-ready graphics that are easy for an audience to interpret.

#create boxplot that displays temperature distribution for each month in the dataset
boxplot(Temp~Month,
data=airquality,
main="Temperature Distribution by Month",

```
xlab="Month",
ylab="Degrees (F)",
col="steelblue",
border="black"
)
```

Executing this syntax generates a chart that clearly separates the temperature distributions for each month. We can visually observe seasonal trends, noting the general increase in the median temperature and the overall spread as the months progress from May into summer. This comparative display is significantly more insightful than viewing five individual plots separately.



Leveraging the Grammar of Graphics with ggplot2

While base R is highly capable, the ggplot2 package offers a more structured, intuitive, and aesthetically pleasing approach to data visualization, based on Leland Wilkinson's Grammar of Graphics. The core principle of ggplot2 is building plots layer by layer, starting with data and aesthetic mappings, followed by geometric objects (geoms).

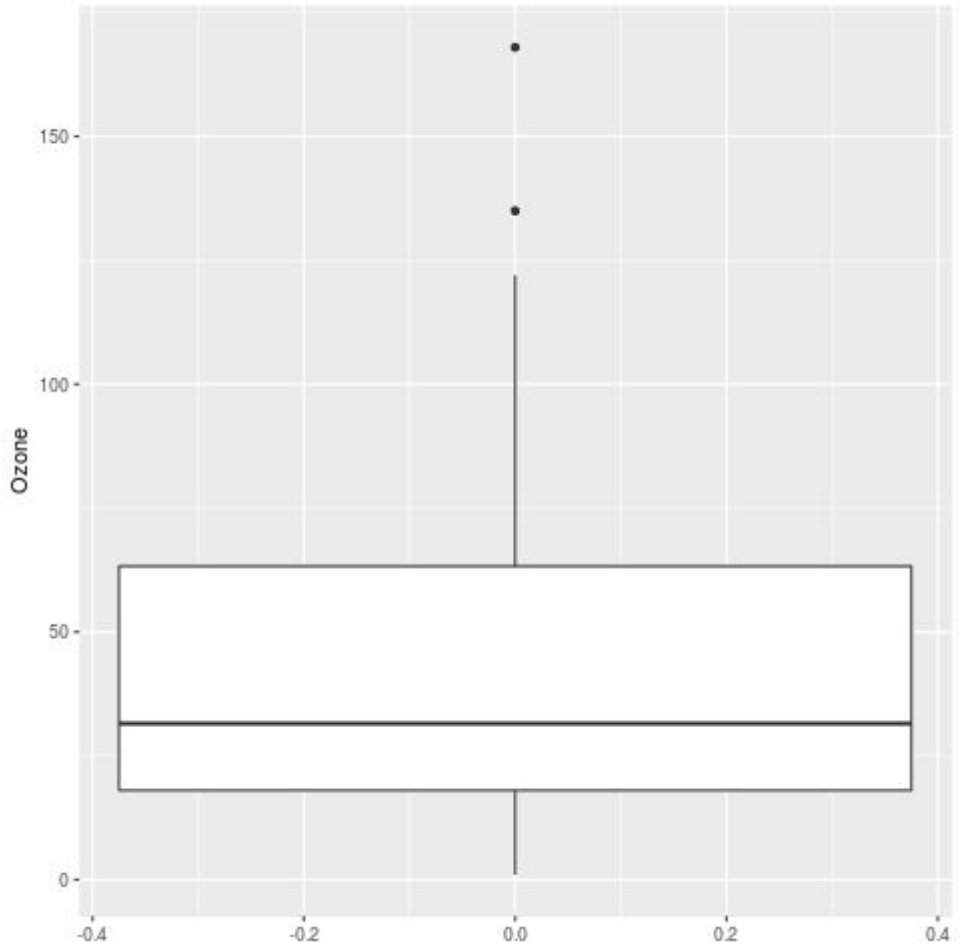
To begin working with ggplot2, the package must first be loaded into the R session using the `library()` function. We will continue to use the **airquality** data frame for consistency. The major difference from base R is how variables are mapped: all variable assignments (which variable goes to the x-axis, y-axis, color, etc.) must be defined within the `aes()` (aesthetic) function.

To replicate the single-variable box plot of the "Ozone" concentration using ggplot2, we first define the data source and map the Ozone variable to the y-axis, as box plots are typically vertical. We then add the geometric layer, `geom_boxplot()`, which instructs ggplot2 to draw the box plot structure.

#create boxplot for the variable "Ozone"
library(ggplot2)

ggplot(data = airquality, aes(y=Ozone)) + geom_boxplot()

This code yields a box plot visually similar to the one created by base R, but it establishes the foundational syntax for all subsequent ggplot2 visualizations. Notice that unlike base R, the variable is mapped to an aesthetic (`y=Ozone`) within the `aes()` call, rather than passed directly as the primary argument.



Generating Comparative Boxplots with ggplot2

When aiming to generate multiple box plots--one for each category--ggplot2 requires mapping both the continuous variable (Y) and the grouping variable (X) within the `aes()` function. As the "Month" variable is coded numerically (5 through 9), it is often necessary to explicitly treat it as a discrete, categorical factor on the x-axis. This is usually accomplished by wrapping the variable in `as.character()` or `as.factor()` within the aesthetic mapping.

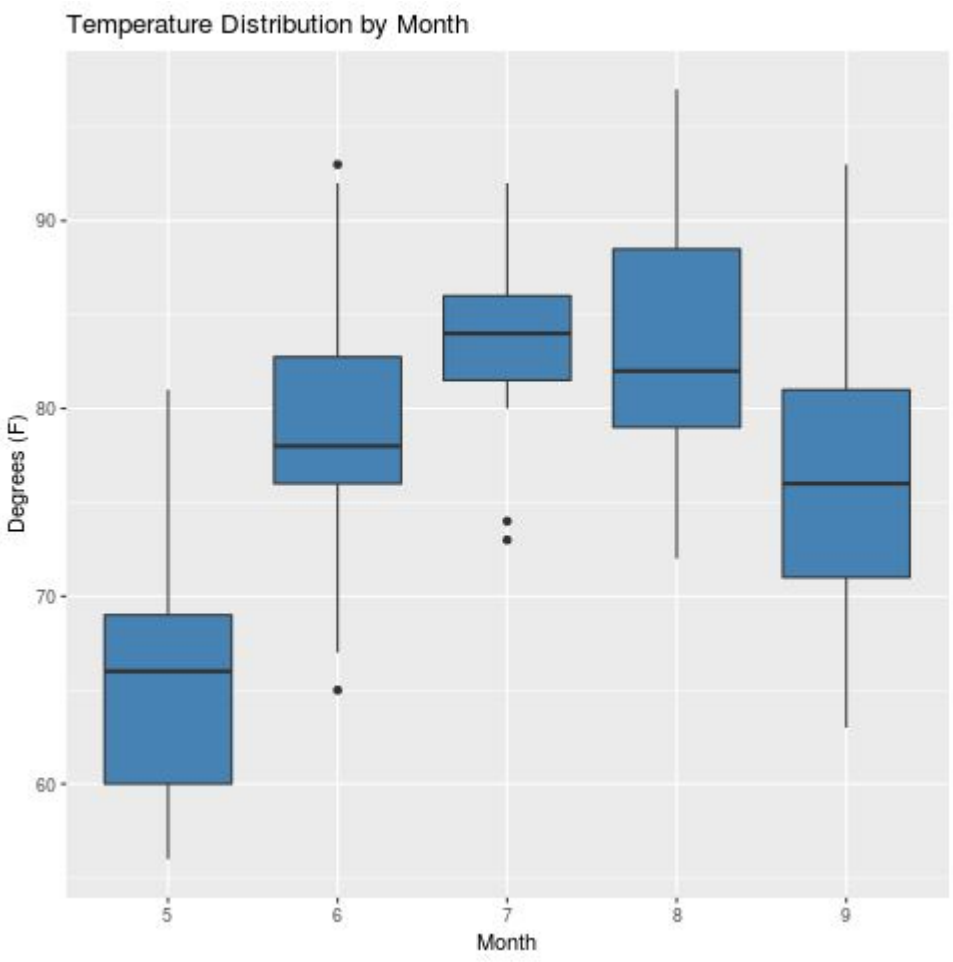
We want to display the distribution of **Temp** (Y) grouped by **Month** (X). By mapping the month to the x-axis and the temperature to the y-axis, and then applying `geom_boxplot()`, ggplot2 automatically handles the grouping and plots the distributions side-by-side.

Customization in ggplot2 is handled through additional layers. We can add color to the boxes by specifying the `fill` parameter within `geom_boxplot()`. Furthermore, instead of using parameters like `main`, `xlab`, and `ylab` as in base R, ggplot2 uses the dedicated `labs()` function to manage all plot titles and axis labels. This layered approach ensures that the code remains clean and highly readable, making it easy to add complexity without sacrificing clarity.

#create boxplot that displays temperature distribution for each month in the dataset
library(ggplot2)

```
ggplot(data = airquality, aes(x=as.character(Month), y=Temp)) +
geom_boxplot(fill="steelblue") +
labs(title="Temperature Distribution by Month", x="Month", y="Degrees (F)")
```

Executing this syntax produces a highly customized and informative chart. The use of the `as.character()` function ensures that the months are treated as distinct categories on the x-axis, rather than a continuous numerical range. This visualization allows for straightforward comparison of the quartile ranges and the overall spread of temperature across the five-month period, clearly showing the seasonal differences and variability.



Conclusion and Additional Resources

Whether using the simplicity of base R or the detailed control offered by ggplot2, plotting multiple

boxplots in a single chart is an essential skill for comparative data analysis in R. This technique allows statisticians and data scientists to quickly assess how key statistical measures--such as the median, spread, and presence of outliers--differ across various subgroups within a dataset. Mastering both methods ensures flexibility in visualization depending on the required level of customization and the complexity of the data structure.

For those seeking to further refine their visualization skills or delve deeper into the statistical principles underpinning box plots, the following resources are recommended. These tutorials provide comprehensive details on advanced formatting and statistical interpretation.

The following tutorials offer additional information about box plots: