Plot Categorical Data in R (With Examples)

Authored by

Mohammed looti

November 3, 2025

In the realm of data science and statistical analysis, mastering the visualization of **categorical data** (often referred to as qualitative data) is absolutely essential. Unlike numerical data, categorical data represents observations that fall into discrete groups or labels, such as names, types, or categories. Effectively understanding and communicating the structure of this data type forms the backbone of robust exploratory analysis, particularly when leveraging the powerful capabilities of the **R** statistical programming environment.

These variables categorize observations into distinct levels, making the initial step of any analysis the accurate identification of which variables within a dataset are indeed categorical. This identification dictates the subsequent choice of graphical representation, ensuring that the visualization accurately reflects the underlying distribution and relationships for both exploration and formal communication.

Typical examples of variables that utilize categorical measurement include:

Socioeconomic Status (e.g., "high income," "middle income," "low income")
Product Type ("electronics," "apparel," "groceries")
Survey Responses (e.g., "agree," "neutral," "disagree")

Specialized plotting techniques are necessary to accurately illustrate the **frequency** distribution or the complex relationships inherent in these variables. The selection of the appropriate visualization tool hinges entirely upon the nature and number of variables under examination: whether the focus is on a single categorical variable, the joint distribution of two categorical variables, or the comparison of a continuous numeric variable across different categorical levels.

Visualizing Categorical Data in R

The **R** programming environment is exceptionally well-equipped for generating high-quality visualizations. Its primary strength in this domain comes from the widely adopted **ggplot2** package, which implements the Grammar of Graphics concept. This package allows analysts to produce highly detailed, customized, and publication-ready graphs specifically tailored for various types of data, including complex categorical structures. Crucially, choosing the appropriate visualization method is paramount for ensuring that the distribution, magnitude, and relationships within the data are conveyed without distortion or ambiguity.

To cover the most common analytical scenarios involving qualitative data, we focus on three highly effective plot types:

Bar Charts: These charts are the standard choice for univariate categorical analysis, clearly displaying the absolute counts or relative proportions of observations belonging to each level of a single variable.

Grouped Boxplots: These are powerful comparative tools, ideal for examining how the summary statistics (median, quartiles, range) of a continuous numeric variable vary when segmented by the levels of a categorical grouping factor.
Mosaic Plots: These specialized visualizations are designed to handle multivariate categorical data, showing the joint distribution and dependencies between two or more categorical factors simultaneously.

The subsequent sections will provide practical, step-by-step demonstrations of how to construct and interpret each of these critical visualizations using **R** and the **ggplot2** package. For continuity across all examples, we will utilize a small, consistent sports-related **data frame** that contains both categorical (team, result) and numeric (points, rebounds) variables.

Example 1: Creating and Customizing Bar Charts

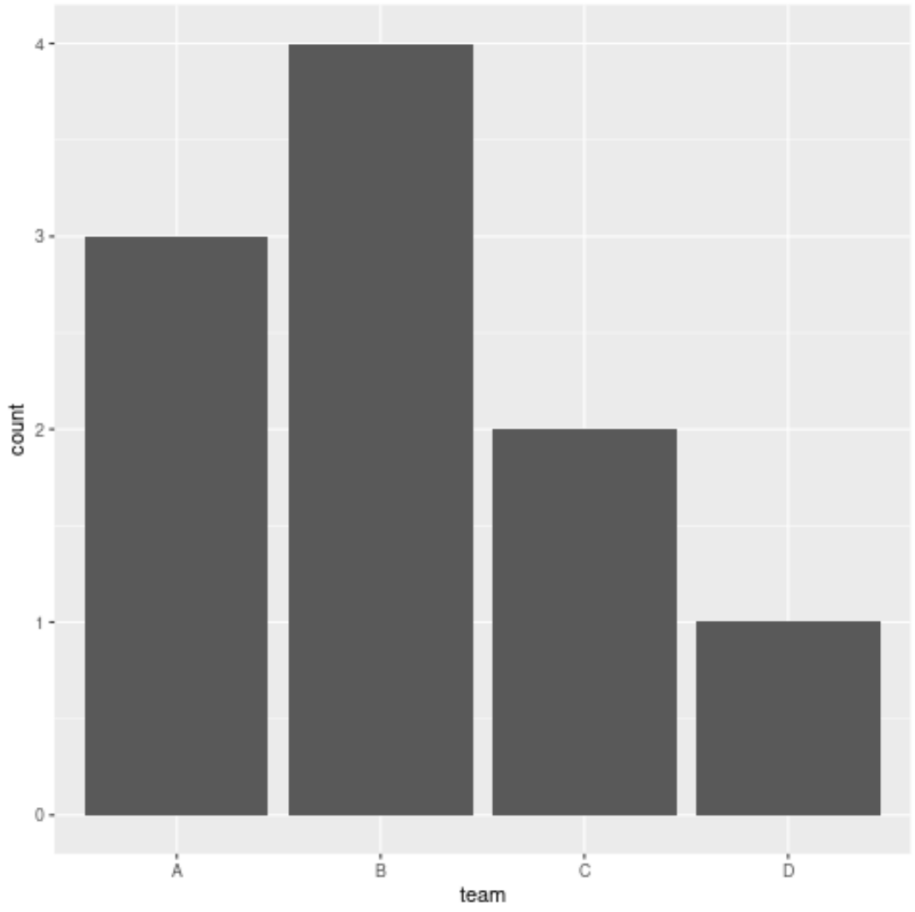
The bar chart remains the cornerstone visualization for analyzing a single **categorical data** variable. Its simplicity is its strength: it visualizes the count or proportion of observations within each category using rectangular bars whose length is directly proportional to the recorded count. This method provides an immediate, intuitive understanding of the variable's distribution.

Our initial demonstration utilizes the **ggplot2** package to generate a fundamental bar chart. This code snippet establishes our sample **data frame** and then maps the categorical variable 'team' to the x-axis, allowing `geom_bar()` to automatically calculate and plot the raw counts for each team entry.

library(ggplot2)

#create data frame
df <- data.frame(result = c('W', 'L', 'W', 'W', 'W', 'L', 'W', 'L', 'W', 'L'),
team = c('B', 'B', 'B', 'B', 'D', 'A', 'A', 'A', 'C', 'C'),
points = c(12, 28, 19, 22, 32, 45, 22, 28, 13, 19),
rebounds = c(5, 7, 7, 12, 11, 4, 10, 7, 8, 8))

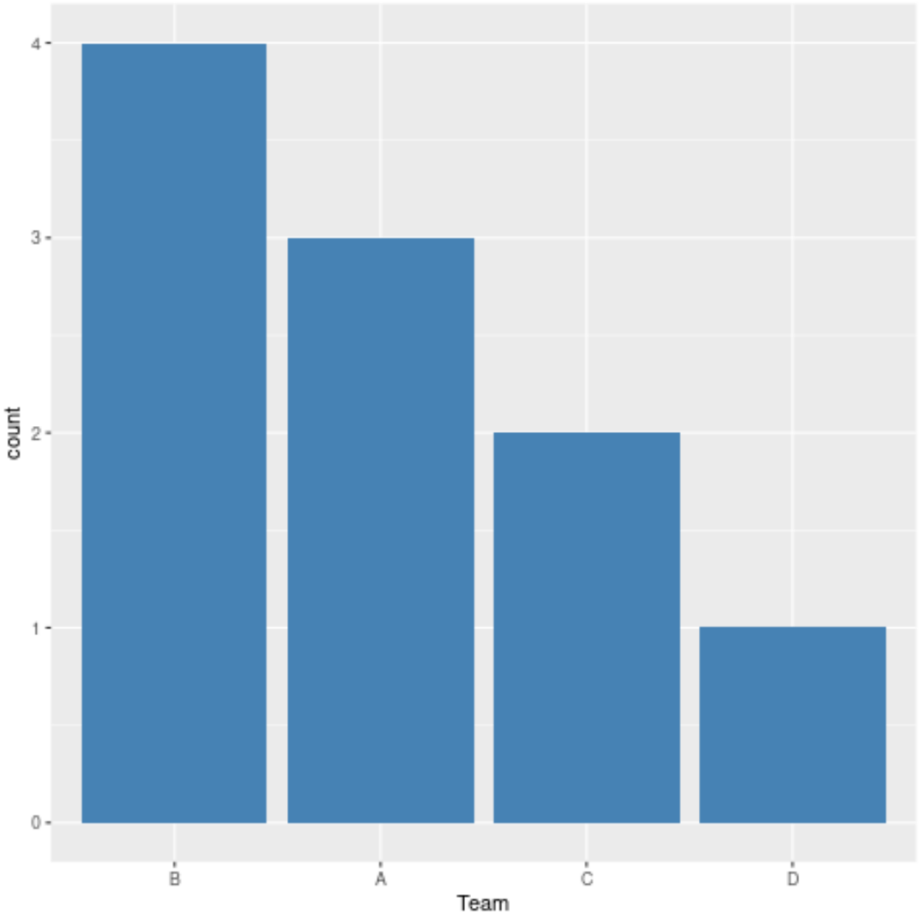
#create basic bar chart of teams
ggplot(df, aes(x=team)) +
geom_bar()



Upon viewing the initial output, the x-axis clearly delineates the four unique categorical levels (Team A, B, C, D), while the y-axis quantifies the raw **frequency** of observations corresponding to each team within the dataset. This visualization immediately highlights which team has the highest representation or activity count in our sample data.

While the initial chart is functional, enhancing readability is crucial for professional presentation. A common best practice is to order the bars based on their magnitude, typically in descending order. This strategic customization allows the viewer to instantly recognize the hierarchy of categories without needing to visually scan all bars. We implement this ordering by incorporating the `reorder()` function directly within the aesthetic mapping of our **ggplot2** call, combined with a function that calculates the negative length (count) of the variable to ensure descending order.

```
#create bar chart of teams, ordered from large to small
ggplot(df, aes(x=reorder(team, team, function(x)-length(x)))) +
geom_bar(fill='steelblue') +
labs(x='Team')
```



The resulting ordered chart is significantly more informative. Furthermore, we have applied cosmetic enhancements, including a distinct fill color and a precise label for the x-axis, confirming the unparalleled flexibility of **ggplot2** for generating high-quality visualizations suitable for any analytical report or publication.

Example 2: Analyzing Distributions with Grouped Boxplots

While bar charts excel at summarizing counts, **boxplots** serve a fundamentally different purpose: visualizing the five-number summary and distribution of continuous (numeric) data. The true power of this method emerges when we create grouped boxplots, which allow us to overlay the numeric distribution across the distinct levels of a **categorical variable**. This technique is indispensable in exploratory data analysis and initial hypothesis testing, enabling analysts to quickly identify shifts in central tendency and variability between groups.

Grouped boxplots offer an insightful visual comparison of key distributional characteristics--including the median, the spread of the middle 50% of the data (Interquartile Range or IQR), and the presence of extreme values (outliers)--all conditioned on the categorical factor. By setting the categorical variable on the x-axis and the numeric variable on the y-axis, we generate side-by-side

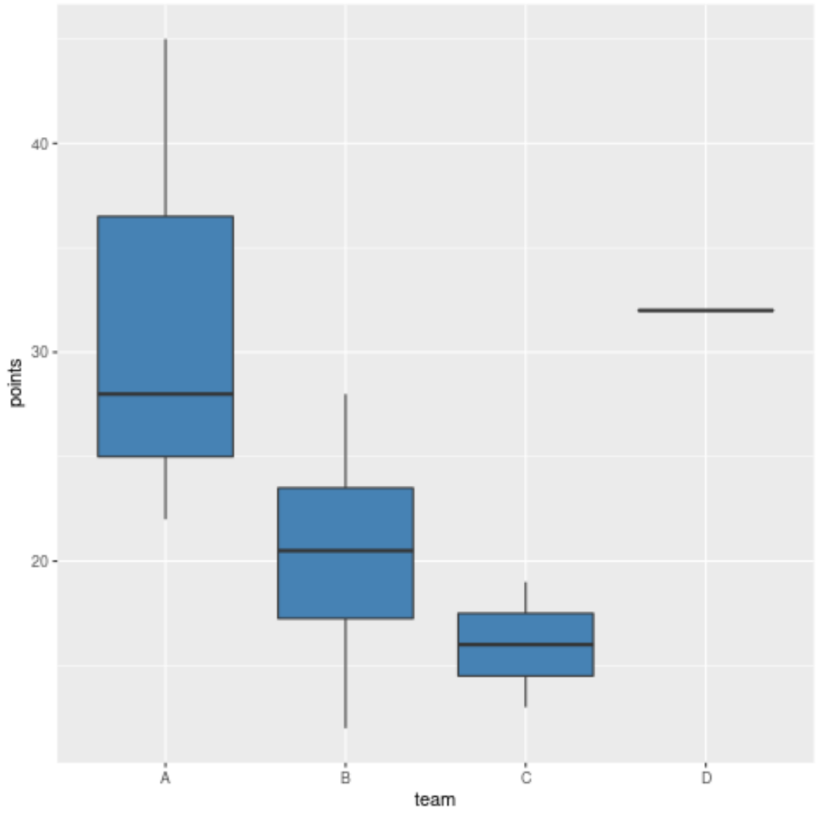
summaries that facilitate direct performance or characteristic comparisons.

The following **R** code demonstrates how to plot the distribution of 'points scored' (our numeric variable) across the levels of 'team' (our categorical variable). We utilize `geom_boxplot()` within **ggplot2** to achieve this grouping.

library(ggplot2)

```
#create data frame
df <- data.frame(result = c('W', 'L', 'W', 'W', 'W', 'L', 'W', 'L', 'W', 'L'),
team = c('B', 'B', 'B', 'B', 'D', 'A', 'A', 'A', 'C', 'C'),
points = c(12, 28, 19, 22, 32, 45, 22, 28, 13, 19),
rebounds = c(5, 7, 7, 12, 11, 4, 10, 7, 8, 8))

#create boxplots of points, grouped by team
ggplot(df, aes(x=team, y=points)) +
geom_boxplot(fill='steelblue')
```



The resulting visual summary allows for immediate cross-group comparison. The horizontal line within each box represents the group median, providing a clear reference point for central

tendency. By analyzing the vertical length of the boxes and whiskers, we can assess the variability and overall range of points scored for each team. Any points plotted outside the whiskers indicate potential outliers. This visualization immediately reveals performance discrepancies and differences in score consistency across the various teams.

Example 3: Exploring Relationships with Mosaic Plots

When the analysis shifts to examining the relationship between two or more **categorical variables**, visualizations must move beyond simple counts. The **mosaic plot**, sometimes known as a Marimekko chart, is the specialized solution for depicting the joint distribution of multiple factors. It presents the frequencies of category combinations in a single, partitioned rectangular space, making it a powerful tool for multivariate qualitative analysis.

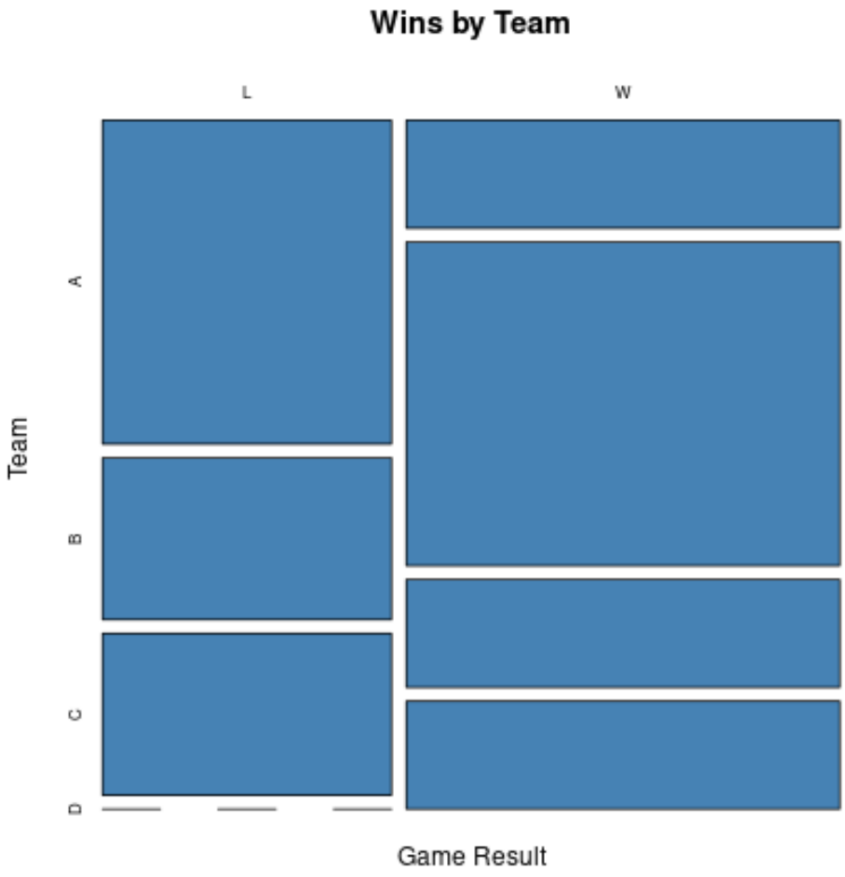
The fundamental principle of the mosaic plot is that the area of each rectangular tile corresponds directly to the proportion of observations that share that specific combination of categories. This makes it particularly effective for visually representing a contingency table and assessing potential dependencies or independence between the variables. If the variables are independent, the conditional frequencies (the vertical segments) should be roughly equal across the marginal frequencies (the horizontal segments).

The code below illustrates the construction of a mosaic plot to visualize the joint **frequency** of the 'result' (Win/Loss) and 'team' variables. Note that for this visualization, we typically rely on base **R** plotting functions, such as `mosaicplot()`, after first generating a cross-tabulation table of counts.

```
#create data frame
df <- data.frame(result = c('W', 'L', 'W', 'W', 'W', 'L', 'W', 'L', 'W', 'L'),
team = c('B', 'B', 'B', 'B', 'D', 'A', 'A', 'A', 'C', 'C'),
points = c(12, 28, 19, 22, 32, 45, 22, 28, 13, 19),
rebounds = c(5, 7, 7, 12, 11, 4, 10, 7, 8, 8))

#create table of counts
counts <- table(df$result, df$team)

#create mosaic plot
mosaicplot(counts, xlab='Game Result', ylab='Team',
main='Wins by Team', col='steelblue')
```



In this resulting visualization, the total width of the plot is segmented by the marginal **frequency** of the first variable ('result'). Within those segments, the vertical divisions show the conditional frequencies of the second variable ('team'). This structure provides an immediate visual diagnostic of how game results are distributed across the four teams, allowing analysts to quickly identify potential patterns, associations, or unexpected deviations in the combined categorical data.

Best Practices for Categorical Visualization

Creating impactful visualizations in the **R** environment extends beyond merely executing plotting commands; it necessitates strict adherence to established data visualization best practices. When working with **categorical data**, ambiguities often arise if plots are poorly structured, inadequately labeled, or rely on default settings, potentially leading to significant misinterpretation of the underlying data patterns.

A primary consideration is the logical ordering of categorical levels. As demonstrated in our bar chart example, ordering categories by their **frequency** (e.g., in descending count) dramatically improves the plot's readability and allows for immediate identification of the most dominant categories. If the variable is ordinal--meaning categories have an inherent rank, such as 'low,' 'medium,' and 'high'--it is essential to preserve this natural hierarchy on the axis rather than sorting

alphabetically or by count. For nominal variables, sorting by frequency is generally the most effective strategy.

Strategic use of color is another fundamental principle. In the preceding examples, a single hue was sufficient because the focus was either on a single count variable or on comparing a numeric variable across groups. However, when introducing a third categorical dimension (e.g., using color to represent the 'result' category in a bar chart of 'team'), careful color selection is required. Always prioritize colorblind-friendly palettes and ensure that the color mapping is clearly and explicitly defined through an accompanying legend to maintain clarity and accessibility.

Lastly, the importance of clear metadata--specifically, descriptive axis labels and a concise title--cannot be overstated. Default labels generated by tools like **ggplot2** or base **R** functions often lack context necessary for a professional audience. Taking the time to customize these elements, as we did with the `labs()` function, guarantees that the audience immediately understands the variables being measured, the units of measurement, and the overarching context of the visualization.

Conclusion and Further Learning

The effective visualization of qualitative data is an indispensable component of any modern data analysis workflow. As we have demonstrated, whether your objective is to quantify the marginal distribution of a single factor via a bar chart, conduct a comparative distributional analysis of a numeric variable across grouped **boxplots**, or investigate complex joint relationships using a **mosaic plot**, the R ecosystem offers exceptionally robust and flexible tools for these tasks.

Achieving publication-quality graphics that accurately and engagingly reflect the structure of your data relies heavily on mastering the underlying principles, particularly those implemented by the ggplot2 package. By consistently applying the techniques illustrated in these examples--focusing on appropriate plot selection, logical ordering, and clear labeling--data analysts can successfully transition from basic tabular summaries to generating deep, informative visual insights that drive effective communication and decision-making.

For those seeking to expand their visualization repertoire, consider exploring tutorials on how to generate other essential plot types in R: