

How to Create a Scree Plot in R (Step-by-Step)

Authored by
stats writer

December 6, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Create a Scree Plot in R (Step-by-Step)*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=106546>

Principal components analysis (PCA) is a powerful statistical technique widely used in data science and machine learning for dimensionality reduction. The core objective of **PCA** is to transform a high-dimensional dataset into a lower-dimensional one while retaining most of the variability present in the original data. This transformation is achieved by identifying orthogonal linear combinations of the original predictor variables--known as **principal components**--that sequentially explain the maximum possible variance.

A critical step following the execution of PCA is determining the optimal number of principal components to retain for subsequent analysis. If too few components are kept, valuable information may be lost; if too many are retained, the benefits of dimensionality reduction are nullified. One of the easiest and most effective ways to visualize the percentage of total variation explained by each component is through the creation of a **scree plot**.

A scree plot is essentially a line chart that displays the eigenvalues (or variance explained) associated with each principal component in descending order. By visualizing this drop-off in explained variance, analysts can quickly apply criteria, such as the famous "elbow rule," to make an informed decision about component selection. This comprehensive tutorial provides a detailed, step-by-step guide on how to perform PCA and generate a clean, informative scree plot using the R programming language.

Understanding Principal Component Analysis (PCA)

Before diving into the R implementation, it is essential to solidify the theoretical foundation of PCA. PCA works by projecting the data onto a new set of coordinates (the principal components) such that the variance along these new axes is maximized. These new axes are uncorrelated, meaning they capture unique dimensions of variability.

When we perform PCA on a dataset, we are fundamentally interested in understanding how much of the total variability within the dataset can be attributed to each resulting principal component. For instance, the first principal component (PC1) will always capture the largest possible amount of variance, PC2 captures the second largest amount of remaining variance, and so forth. The collective variance explained by the first few components often accounts for over 80% or 90% of the dataset's total information.

The mathematical backbone of PCA relies on calculating the eigenvalues of the covariance or correlation matrix of the input data. Each eigenvalue corresponds directly to the amount of variance explained by its respective principal component. Plotting these eigenvalues in decreasing order gives us the visual representation known as the scree plot.

The Importance of Eigenvalues and Variance Explained

The decision of how many principal components to retain is critical for successful dimensionality reduction. If a component has a large eigenvalue, it means that component captures a significant portion of the data's variability. Conversely, components with small eigenvalues contribute very little unique information and are generally discarded.

The total variance in the dataset is the sum of all eigenvalues. The proportion of variance explained by a single principal component is calculated by dividing its eigenvalue by the sum of all eigenvalues. This proportion is what the vertical axis of the scree plot represents. By visualizing this decline, we can apply two main criteria for selection:

The Kaiser Criterion: This rule suggests retaining only those principal components whose corresponding eigenvalues are greater than 1. Since standardized data typically has a total variance equal to the number of variables (and thus, an average eigenvalue of 1), any component explaining less variance than a single variable contributes minimally to the overall structure.

The Elbow Rule: This is the visual criterion utilized by the scree plot. We look for a point on the plot where the smooth, steep curve of large eigenvalues bends abruptly into a flat, trailing line. This "elbow" point indicates where the marginal gain in explained variance significantly decreases, suggesting that subsequent components contribute little value.

Step 1: Preparing the Dataset in R

To demonstrate the creation of a scree plot, we will utilize the built-in R dataset called `USArrests`. This dataset is excellent for demonstration as it contains data on arrests per 100,000 residents in each U.S. state for four types of crimes: Murder, Assault, UrbanPop (percentage urban population), and Rape.

The first step in any data analysis workflow in R is to load the necessary data and inspect its structure. This ensures the data is correctly formatted and allows us to verify the variable types before proceeding with statistical modeling. Since `USArrests` is a default dataset, it only requires a simple call to load it into the environment.

The following code snippet loads the data and displays the first six observations, confirming the structure of our variables:

```
# Load the internal USArrests dataset  
data("USArrests")  
  
# Display the initial structure and first six rows of the data  
head(USArrests)
```

	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7

Step 2: Executing Principal Component Analysis (PCA)

With the data loaded, the next step is to execute the PCA. In R, the standard function for this task is the prcomp() function, which is part of the base R installation (specifically, the `stats` package). This function calculates the components based on the singular value decomposition (SVD) of the data matrix, which is highly stable for computation.

A critical parameter in the prcomp() function is `scale = TRUE`. Because the variables in the `USArrests` dataset are measured on wildly different scales (e.g., Assault numbers are in the hundreds, while Murder rates are low single digits or teens), it is mandatory to standardize the data before running PCA. Setting `scale = TRUE` ensures that all variables are transformed to have a mean of zero and a standard deviation of one prior to component extraction, preventing larger-magnitude variables from dominating the results.

The result of the PCA is stored in the variable `results`, which is a list object containing several components necessary for plotting, including the standard deviations (`sdev`) of the principal components.

```
# Perform PCA using the prcomp() function.  
# Scaling is crucial since the variables are on different units.  
results <- prcomp(USArrests, scale = TRUE)
```

Step 3: Calculating and Analyzing Explained Variance

The `results` object generated in Step 2 contains the standard deviations (`sdev`) of the principal components. These standard deviations are the square root of the eigenvalues. To determine the variance explained by each component, we must first square the standard deviations to get the eigenvalues, and then calculate the proportion of the total variance captured by each component.

The proportion of variance explained by a single principal component is calculated by dividing its eigenvalue by the sum of all eigenvalues. This proportion is the core metric displayed on the vertical axis of the scree plot.

The R code below performs this necessary calculation, storing the proportions in the `var_explained` vector. This vector is the essential input for generating the visualization in the subsequent step.

```
# Square the standard deviations (sdev) to obtain the eigenvalues (variance)  
# Then divide each eigenvalue by the total sum of eigenvalues to get the proportion of  
variance explained.  
var_explained = results$sdev^2 / sum(results$sdev^2)
```

For numerical confirmation, we can print the `var_explained` vector to see the exact percentage contribution of each component:

```
print(var_explained)
```

```
0.62006039 0.24744129 0.08914080 0.04335752
```

This numerical breakdown provides clear insight into the data reduction potential:

The first principal component (PC1) explains approximately **62.01%** of the total variation in the dataset.

The second principal component (PC2) explains approximately **24.74%** of the total variation in the dataset.

The third principal component (PC3) explains approximately **8.91%** of the total variation in the dataset.

The fourth principal component (PC4) explains approximately **4.34%** of the total variation in the dataset.

Notice that the sum of these proportions is exactly 1.00 (100%), confirming that all variance from the original four variables is fully represented across the four principal components.

Step 4: Generating the Scree Plot with ggplot2

To visualize the calculated variance proportions, we will use the highly flexible and robust `ggplot2` package. This package adheres to the grammar of graphics, allowing precise control over visualization elements. We use `qplot()` for simplicity, which plots the component index (1 to 4) against the variance explained.

We add `geom_line()` to connect the points, forming the characteristic curve of a scree plot.

Furthermore, we ensure the plot is clearly labeled by defining the X-axis as "Principal Component" and the Y-axis as "Proportion of Variance Explained," and setting the Y-axis limits from 0 to 1 for proper scale representation.

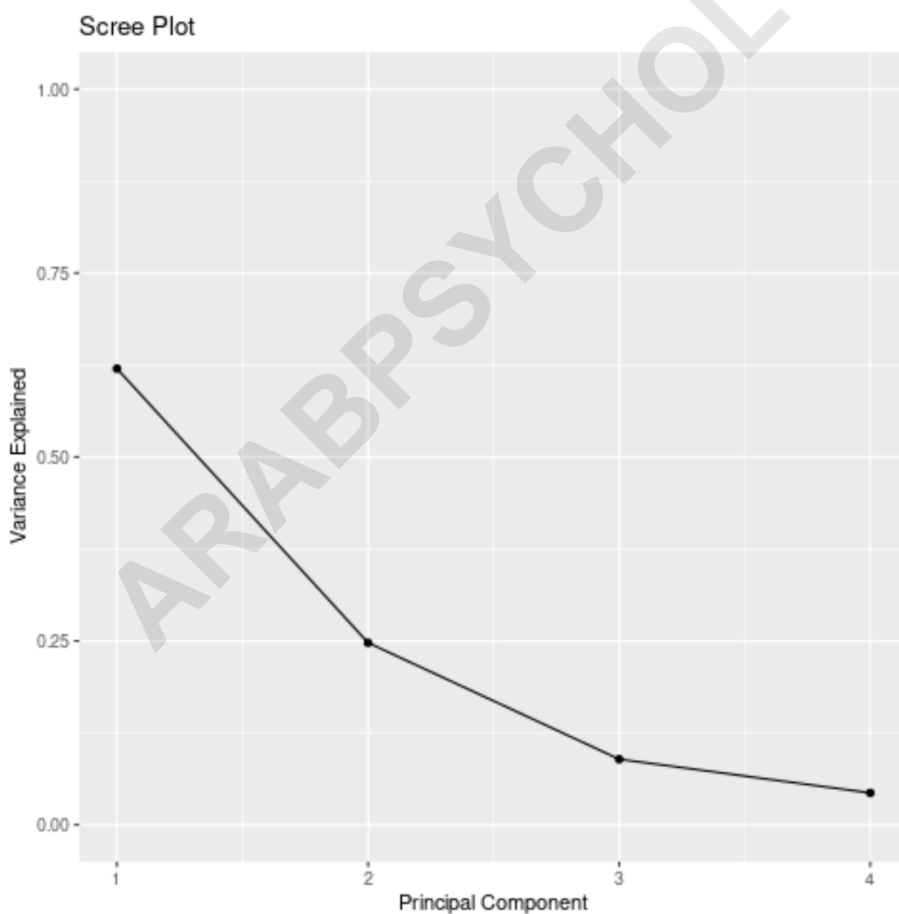
Load the ggplot2 library for visualization

library(ggplot2)

Create the scree plot

```
qplot(c(1:4), var_explained) +  
geom_line() +  
xlab("Principal Component") +  
ylab("Variance Explained") +  
ggtitle("Scree Plot") +  
ylim(0, 1)
```

The resulting graph visually represents the rapid decline in marginal variance gain:



Interpreting the Scree Plot and Component Selection

Interpretation of the scree plot is crucial for concluding the PCA analysis. The plot allows us to visually apply the "Elbow Rule."

The X-axis displays the principal component and the Y-axis displays the percentage of total variance explained by each individual principal component. The plot shows a very steep drop from PC1 to PC2, indicating that PC1 accounts for the overwhelming majority of the information. While PC2 still contributes significantly (24.74%), the curve begins to flatten dramatically after this point, particularly when moving from PC2 to PC3 and PC4.

Applying the Elbow Rule, we identify the "elbow" where the marginal benefit of adding a new component diminishes rapidly. In this visualization, the elbow is clearly located at the transition point to PC3. Therefore, the standard recommendation based on the scree plot is to retain the first **two principal components**. By keeping PC1 and PC2, we successfully reduce the dimensionality from four variables to two, while retaining approximately 86.75% of the original variance.

Advanced Scree Plot Generation via Factoextra

For analysts seeking a more automated and visually rich approach, the `factoextra` package provides streamlined functionality for plotting PCA results. This package leverages `ggplot2` internally but simplifies the code required to generate complex graphs, including the scree plot.

To generate the scree plot using `factoextra`, the user would typically install the package and use the `fviz_eig()` function, often setting the `type` argument to "bar" for a bar chart representation of the eigenvalues, which is often preferred for presentation clarity. This function computes and visualizes the eigenvalues directly from the PCA result object (`results` in our case).

Using `fviz_eig()` not only reduces the complexity of manual calculation and plotting but also allows for easy customization of labels, titles, and even the inclusion or exclusion of specific components, ensuring a clean and effective visualization output tailored to specific analytical needs.

Mastering the creation and interpretation of the scree plot is a fundamental skill for anyone performing Principal components analysis (PCA). By utilizing robust R functions like the `prcomp()` function and powerful visualization tools like `ggplot2`, analysts can efficiently reduce dimensionality while ensuring maximum data fidelity. This ensures that subsequent machine learning models or statistical tests are built on the most informative components, leading to cleaner, more interpretable results.

You can find more machine learning tutorials on our site.