Learning to Visualize Principal Components: A Step-by-Step Guide to Creating Scree Plots in R

Authored by

Mohammed looti

November 5, 2025

The methodology of Principal components analysis (PCA) stands as an indispensable statistical technique, primarily utilized for the critical task of **dimensionality reduction**. In the realm of data science, where datasets often contain numerous highly correlated variables, PCA offers an elegant solution: transforming this complexity into a smaller, more manageable set of linearly uncorrelated variables known as **principal components**. These components are expertly constructed as linear combinations of the original predictor variables, meticulously designed to sequentially capture and explain the maximum possible amount of variance within the dataset. By focusing on variance, PCA efficiently distills the core information contained across the original features.

A central objective when executing PCA is the quantitative assessment of how much total data variation is successfully captured by each resulting principal component. Given that these components are mathematically ordered based on their explanatory power--from PC1 (most variance) down to PCn (least variance)--data practitioners require a clear, intuitive visualization tool. This tool must effectively guide the critical decision regarding component retention, ensuring that the selected subset still represents the data's inherent structure.

The most effective and universally accepted method for visually representing the proportion of variation explained by each principal component is the construction of a **scree plot**. This powerful graphical tool plots the eigenvalues (or explained variance proportions) against the component number. Its primary function is to assist analysts in identifying the "elbow" or inflection point, thereby facilitating a statistically sound decision on how many components should be retained for subsequent modeling, clustering, or visualization tasks, ultimately preventing the inclusion of noise.

This comprehensive tutorial is designed to provide a precise, detailed, and replicable step-by-step methodology for generating a high-quality, interpretable scree plot using the powerful, open-source statistical programming language, R. We will leverage R's robust capabilities for statistical computation and sophisticated visualization via the popular `ggplot2` package.

Step 1: Loading and Preparing the Sample Dataset in R

To demonstrate the fundamental process of creating a scree plot, we will utilize a widely recognized, built-in R dataset known as **USArrests**. This dataset is frequently employed in statistical examples involving multivariate analysis and provides excellent metrics suitable for our initial PCA. Understanding the structure and nature of the data is the prerequisite to any successful statistical analysis.

The **USArrests** dataset contains crucial statistical information regarding the number of arrests per 100,000 residents for four distinct categories of violent crimes across each of the 50 U.S. states in 1973. The four key variables are: **Murder**, **Assault**, **UrbanPop** (percentage of the population living in urban areas), and **Rape**. Recognizing that these variables are measured on vastly different scales (e.g., Assault numbers are much larger than Murder rates) highlights the absolute necessity

for data preprocessing before PCA, a step we will address shortly.

Our initial process begins by loading the dataset into the R environment and conducting a preliminary inspection of the initial rows using the standard `head()` function. This critical step ensures the data integrity, confirms correct loading, and allows us to quickly verify the structural format of the variables, ensuring they are suitable for numerical processing, as illustrated in the following R code block:

 # Load the standard R dataset: USArrests
data("USArrests")

# View the first six observations to confirm structure
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: Performing Principal Components Analysis with Scaling

Once the dataset is loaded and verified, the next crucial phase is the execution of the Principal components analysis itself. In the R programming environment, this complex computation is handled efficiently and robustly using the powerful built-in function, `prcomp()`. This function is the canonical tool for performing PCA on standardized numerical data, calculating the necessary covariance or correlation matrix internally.

A fundamental consideration before running PCA is the need to standardize (or scale) the variables. Standardization is imperative when the input variables are measured on different scales, as is demonstrably the case with the **USArrests** data (e.g., Assault ranges up to 337, while Murder ranges up to 17.4). Without standardization, variables with inherently larger numerical values, such as Assault, would possess disproportionately high variance, allowing them to exert an undue, artificial influence on the resulting principal components, skewing the analysis.

We achieve this vital standardization process by passing the argument `scale = TRUE` within the function call to prcomp(). This instructs R to internally scale each column (variable) to have a mean of zero and a standard deviation of one before the PCA calculation commences. The comprehensive results of the PCA, including the standard deviations, loadings, and rotational

matrices, are then stored in a new R object named `results`, ready for subsequent extraction and visualization:

```
# Perform PCA on the USArrests data, ensuring variables are scaled (standardized)
results <- prcomp(USArrests, scale = TRUE)
```

Step 3: Calculating the Proportion of Explained Variance

Before we can construct the scree plot visualization, we must first perform the necessary mathematical step: calculating the proportion of the total data variance that is uniquely explained by each successive **principal component** derived from the `results` object. This step is crucial because the scree plot itself is a graphical representation of these calculated proportions.

The total variance within a dataset subjected to PCA is mathematically equivalent to the sum of the variances of all individual principal components. In the context of the output provided by the prcomp() function, the variance of a principal component is found by squaring its standard deviation. The standard deviations for all components are conveniently stored within the object component `results$sdev`. These squared standard deviations are equivalent to the **eigenvalues** of the correlation matrix.

To determine the percentage (or proportion) of variance explained by a specific component, we follow a simple ratio calculation: we divide the variance of that individual component by the total variance across all components. Since the data was scaled, the sum of variances (eigenvalues) will equal the number of variables (four, in this case). The following R code executes this core calculation, storing the resulting proportions--which necessarily sum exactly to 1.0--in the new variable `var_explained`:

```
# Calculate the variance explained by squaring the standard deviation (sdev)
# and dividing by the total variance (sum of squared sdevs)
var_explained = results$sdev^2 / sum(results$sdev^2)
```

Step 4: Generating the Scree Plot Visualization using ggplot2

With the explained variance ratios successfully calculated, we now move to the visualization stage: creating the scree plot itself. This visualization is most effectively generated using the popular R package ggplot2, which is renowned within the R community for its ability to produce highly customizable, publication-quality graphics based on the grammar of graphics philosophy. The clarity and precision offered by `ggplot2` are essential for accurate data presentation.

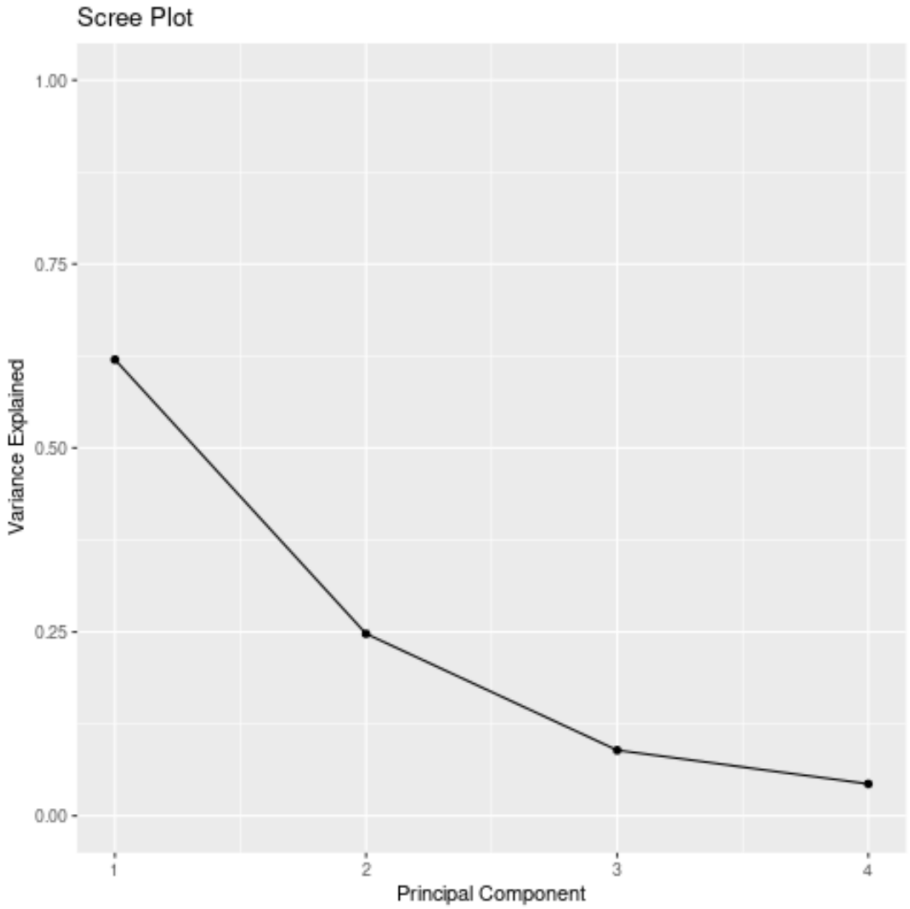
The scree plot's structure is simple yet profoundly informative: it maps the sequential **principal**

component number on the horizontal (x) axis against the proportion of variance explained (or the corresponding eigenvalue) on the vertical (y) axis. The resulting line plot visually displays the explanatory power of the components in descending order. The primary goal of this plot is to identify the critical "elbow" point--the point where the slope of the variance explained curve significantly flattens, signaling diminishing returns for retaining subsequent components.

For rapid, effective plotting, we utilize the `qplot()` function from the `ggplot2` library. We ensure the visualization includes descriptive labels for the axes, a clear title, and explicitly set the y-axis limit to range from 0 to 1.0 (representing 0% to 100% of the total explained variance). This standardized presentation ensures maximum readability and interpretability:

 # Load the visualization library
library(ggplot2)

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



Step 5: Interpreting the Scree Plot and Determining Component Retention

The resulting graph provides an immediate and powerful visual basis for interpretation. The steep drop-off between the first and second components, followed by a much gentler, almost linear slope thereafter, is the key feature we seek. The x-axis confirms the sequential order of the **principal component** (PC) number, while the y-axis presents the corresponding proportion of total variance explained. This pattern confirms that the initial components successfully capture the vast majority of the data's inherent variability, while subsequent components contribute minimally, mostly reflecting noise rather than structure.

Two primary rules guide the interpretation of a scree plot and the decision of component retention: the **Kaiser Criterion** and the **Elbow Rule**. The Elbow Rule suggests retaining all components that precede the point where the curve begins to flatten significantly (the elbow). Observing the plot above, the sharpest decline occurs after PC1 and the slope noticeably flattens after PC2. This visual evidence strongly suggests retaining two principal components. Furthermore, the Kaiser Criterion recommends retaining only those components whose eigenvalues (explained variance before normalization) are greater than 1.0. Given that we have four variables, only the first two

components are likely to exceed this threshold.

To reinforce the visual interpretation with quantitative accuracy, we must examine the exact numerical percentages stored in the `var_explained` vector calculated in Step 3. Printing this vector provides the precise contribution of each component to the total data variation, allowing for a rigorous, data-driven decision:

print(var_explained)

0.62006039 0.24744129 0.08914080 0.04335752

Based on both the comprehensive scree plot visualization and the precise numerical output, we can draw the following crucial conclusions regarding the explanatory power derived from the USArrests dataset:

The first **principal component** (PC1) explains a substantial **62.01%** of the total variation present in the data. This component captures the overwhelming majority of the shared variability among the crime rates.
The second **principal component** (PC2) contributes an additional **24.74%** of the total variation, representing a secondary, but still highly significant, axis of variability.
The third **principal component** (PC3) explains only **8.91%** of the total variation, a contribution that is rapidly diminishing.
The fourth **principal component** (PC4) accounts for the remaining **4.34%** of the total variation, which is negligible for practical purposes.

It is a defining characteristic of the Principal components analysis methodology that the sum of all these proportional variances must equate precisely to 100%, or 1.0 when expressed as a proportion. Critically, in this specific analysis, the first two principal components together capture over 86.75% of the total variability within the violent crime data. This robust finding strongly suggests that only these two components are necessary to retain the core information for most downstream analytical tasks, allowing for effective dimensionality reduction from four variables down to two without significant loss of information.

For further resources, advanced statistical methods, and detailed guides on multivariate analysis and machine learning tutorials in R, please refer to our main page.