Statistics. Simplified.
STATOLOGY

MENU

MARCH 11, 2021 BY ZACH

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

Principal components analysis (PCA) is an unsupervised machine learning technique that seeks to find principal components – linear combinations of the predictor variables – that explain a large portion of the variation in a dataset.

When we perform PCA, we're often interested in understanding what percentage of the total variation in the dataset can be explained by each principal component.

One of the easiest ways to visualize the percentage of variation explained by each principal component is to create a **scree plot**.

This tutorial provides a step-by-step example of how to create a scree plot in R.

Step 1: Load the Dataset

For this example we'll use a dataset called USArrests, which contains data on the number of arrests per 100,000 residents in each U.S. state in 1973 for various crimes.

The following code shows how to load and view the first few rows of this dataset:

```
#load data
data("USArrests")

#view first six rows of 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: Perform PCA

Next, we'll use the **prcomp()** function built into R to perform principal components analysis.

```
#perform PCA
results <- prcomp(USArrests, scale = TRUE)
```

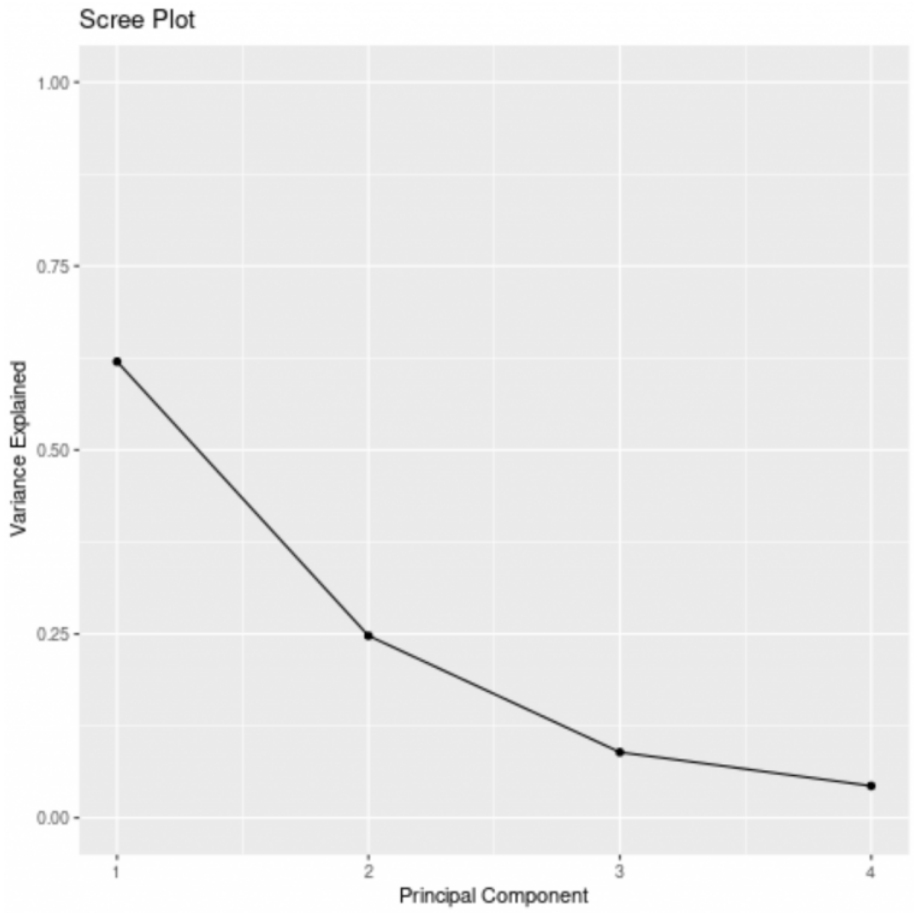
Step 3: Create the Scree Plot

Lastly, we'll calculate the percentage of total variance explained by each principal component and use **ggplot2** to create a scree plot:

```
#calculate total variance explained by each principal component
var_explained = results$sdev^2 / sum(results$sdev^2)

#create scree plot
library(ggplot2)
```

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



The x-axis displays the principal component and the y-axis displays the percentage of total variance explained by each individual principal component.

We can also use the following code to display the exact percentage of total variance explained by each principal component:

```
print(var_explained)

[1] 0.62006039 0.24744129 0.08914080 0.04335752
```

We can see:

- The first principal component explains **62.01%** of the total variation in the dataset.
- The second principal component explains **24.74%** of the total variation in the dataset.
- The third principal component explains **8.91%** of the total variation in the dataset.
- The fourth principal component explains **4.34%** of the total variation in the dataset.

Notice that all of the percentages sum to 100%.

You can find more machine learning tutorials on this page.

Published by Zach

View all posts by Zach

PREV

How to Concatenate Arrays in Python (With Examples)

NEXT

How to Compare Two Columns in Pandas (With Examples)

Leave a Reply