

# PCA Optimization and Reconstruction Error Analysis

## OBJECTIVE

To optimize PCA preprocessing parameters (centering and scaling) and evaluate reconstruction error on a sample dataset.

## LIBRARIES REQUIRED

- **stats** – For PCA (**prcomp**)
- **caret** – For creating parameter grid

## STEPS

1. Create a sample dataset with 3 variables
2. Define grid of PCA parameters (**center**, **scale**.)
3. Iterate over grid to compute RMSE for each config
4. Select best parameter set (lowest RMSE)
5. Perform PCA using optimal settings
6. Visualize using scree plot and biplot
7. Project data, reconstruct original, and compute reconstruction error

## CODE SNIPPET

```
install.packages("caret", repos = "https://cloud.r-project.org")
```

```
## package 'caret' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
## C:\Users\Arun Santhosh R A\AppData\Local\Temp\RtmpqAMjdf\downloaded_packages
```

```

library(stats)
library(caret)

# Create a sample dataset
data <- data.frame(
  x1 = c(1, 2, 3, 4, 5),
  x2 = c(5, 4, 3, 2, 1),
  x3 = c(2, 3, 1, 5, 4)
)

# Define the hyperparameter grid for PCA
param_grid <- expand.grid(
  center = c(TRUE, FALSE),
  scale. = c(TRUE, FALSE)
)

# Initialize variables to store best hyperparameters and best RMSE
best_params <- NULL
best_rmse <- Inf

# Iterate over hyperparameter grid
for (i in 1:nrow(param_grid)) {
  center <- param_grid$center[i]
  scale <- param_grid$scale.[i]

  # Perform PCA
  pca_result <- prcomp(data, center = center, scale. = scale)

  # Calculate RMSE (example metric)
  rmse <- sqrt(sum(pca_result$sdev^2))

  # Update best hyperparameters if RMSE is lower
  if (rmse < best_rmse) {
    best_rmse <- rmse
    best_params <- c(center, scale)
  }
}

# Print the best hyperparameters
print(best_params)

```

```
## [1] FALSE TRUE
```

```

# Perform PCA with the best hyperparameters
pca_result <- prcomp(data, center = best_params[1], scale. = best_params[2])

# View the PCA summary
print(summary(pca_result))

```

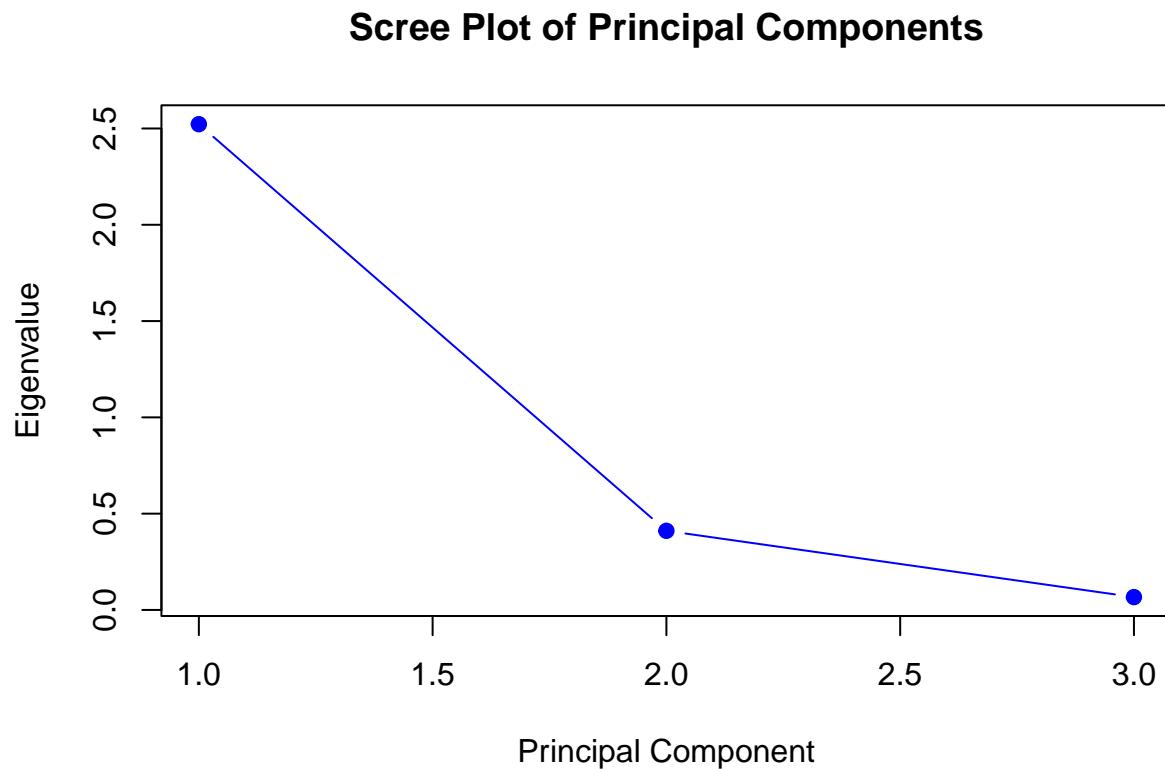
```

## Importance of components:
##              PC1      PC2      PC3
## Standard deviation  1.5882 0.6410 0.25847

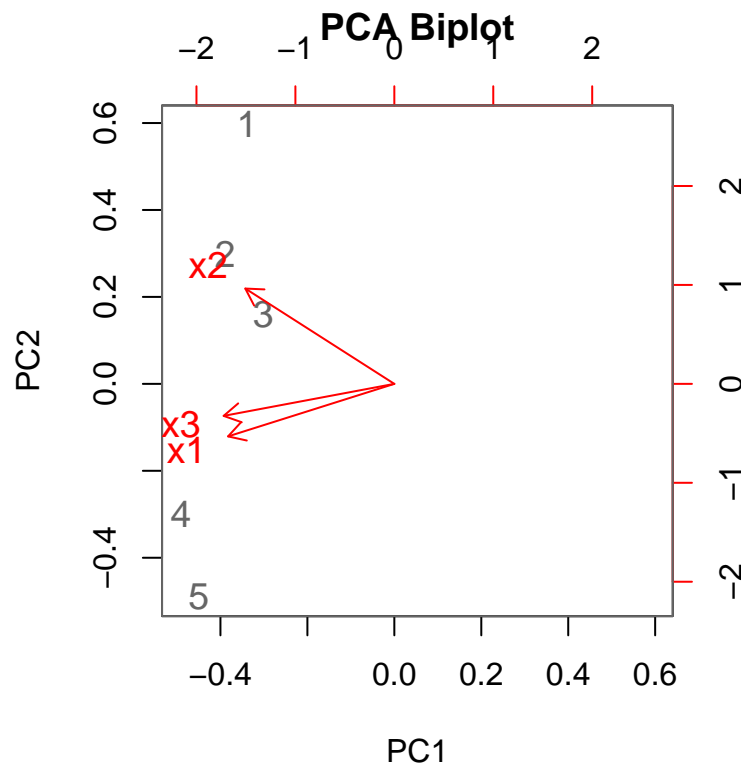
```

```
## Proportion of Variance 0.8408 0.1370 0.02227
## Cumulative Proportion 0.8408 0.9777 1.00000
```

```
# Scree plot of eigenvalues
eigenvalues <- pca_result$sdev^2
plot(eigenvalues, type = "b",
     main = "Scree Plot of Principal Components",
     xlab = "Principal Component",
     ylab = "Eigenvalue",
     col = "blue", pch = 19)
```



```
# PCA biplot
biplot(pca_result,
      cex = 1.2,
      col = c("gray40", "red")) # gray for points, red for vectors
title(main = "PCA Biplot")
```



```
# Project the data onto principal components
projected <- predict(pca_result)

# Reconstruct the original data
reconstructed <- projected %*% t(pca_result$rotation)

# Add back centering if applied
if (!is.null(pca_result$center)) {
  reconstructed <- sweep(reconstructed, 2, pca_result$center, "+")
}

# Rescale if scaling was applied
if (!is.null(pca_result$scale)) {
  reconstructed <- sweep(reconstructed, 2, pca_result$scale, "*")
}

# Compute reconstruction error (MSE)
reconstruction_error <- mean((as.matrix(data) - reconstructed)^2)
cat("Reconstruction Error (MSE):", reconstruction_error, "\n")
```

```
## Reconstruction Error (MSE): 2.830038e-30
```

## CONCLUSION

Optimizing PCA parameters improves dimensionality reduction quality. Reconstruction error quantifies information loss, helping validate the effectiveness of PCA preprocessing.