

# Package: prinsurf (via r-universe)

March 13, 2025

**Title** Constructs Principal Surfaces

**Version** 1.0

**Description** Construct a principal surface that are two-dimensional surfaces that pass through the middle of a p-dimensional data set. They minimise the distance from the data points, and provide a nonlinear summary of data. The surfaces are nonparametric and their shape is suggested by the data. The formation of a surface is found using an iterative procedure which starts with a linear summary, typically with a principal component plane. Each successive iteration is a local average of the p-dimensional points, where an average is based on a projection of a point onto the nonlinear surface of the previous iteration. For more information on principal surfaces, see Ganey, R. 2019, Biplots based on principal surfaces, Faculty of Science ,Department of Statistical Sciences, UCT.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Imports** rgl

**Suggests** stats, Matrix, akima, knitr, rmarkdown

**VignetteBuilder** knitr

**Config/pak/sysreqs** libfreetype6-dev libglu1-mesa-dev make libpng-dev  
libgl1-mesa-dev zlib1g-dev

**Repository** <https://raeesaganey91.r-universe.dev>

**RemoteUrl** <https://github.com/raeesaganey91/prinsurf>

**RemoteRef** HEAD

**RemoteSha** 0472e6bdca32678a442ceb219d6b48b542b9e2e0

## Contents

principal.surface . . . . .	2
-----------------------------	---

---

principal.surface	<i>Principal Surface</i>
-------------------	--------------------------

---

**Description**

A function to compute principal surfaces based on input data containing continuous variables.

**Usage**

```
principal.surface(
  X,
  max.iter = 10,
  alpha = 0.6,
  N = 50,
  print_iterations = FALSE
)
```

**Arguments**

X	A data frame or matrix containing continuous variables.
max.iter	Integer. Maximum number of iterations for the principal surface algorithm.
alpha	Numeric. The span argument passed to the loess() function.
N	Integer. The resolution for the interpolated grid surface, creating an $N^2 \times p$ matrix.
print_iterations	Logical. Should the iterations in the principal surface algorithm be printed? Defaults to FALSE.

**Value**

A list with the following components:

fj.mat	A numeric $n \times p$ matrix of the final principal surface fitted values.
lambda.j	A numeric representation of the samples in two dimensions.

**Examples**

```
## Not run:
surface <- principal.surface(iris[,1:3],max.iter = 3)
## End(Not run)
surface <- principal.surface(iris[1:50,1:3],max.iter = 3)
```

# Index

principal.surface, [2](#)