

# Package ‘GeomArchetypal’

June 24, 2024

**Type** Package

**Title** Finds the Geometrical Archetypal Analysis of a Data Frame

**Version** 1.0.2

**Date** 2024-06-24

**Maintainer** Demetris Christopoulos <dchristop@econ.uoa.gr>

**Description** Performs Geometrical Archetypal Analysis after creating Grid Archetypes which are the Cartesian Product of all minimum, maximum variable values. Since the archetypes are fixed now, we have the ability to compute the convex composition coefficients for all our available data points much faster by using the half part of Principal Convex Hull Archetypal method.

Additionally we can decide to keep as archetypes the closer to the Grid Archetypes ones. Finally the number of archetypes is always 2 to the power of the dimension of our data points if we consider them as a vector space.

Cutler, A., Breiman, L. (1994) <[doi:10.1080/00401706.1994.10485840](https://doi.org/10.1080/00401706.1994.10485840)>.

Morup, M., Hansen, LK. (2012) <[doi:10.1016/j.neucom.2011.06.033](https://doi.org/10.1016/j.neucom.2011.06.033)>.

Christopoulos, DT. (2024) <[doi:10.13140/RG.2.2.14030.88642](https://doi.org/10.13140/RG.2.2.14030.88642)>.

**License** GPL (>= 2)

**Depends** R (>= 3.1.0)

**Imports** Matrix, geometry, archetypal, doParallel, methods, plot3D, distances, rlang, magrittr, dplyr, mirai, abind, scales

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Author** Demetris Christopoulos [aut, cre, cph],  
David Midgley [ctb, cph],  
Sunil Venaik [ctb],  
INSEAD Hoffmann Institute France [fnd],  
The University of Queensland Australia [fnd]

**NeedsCompilation** no

**Language** en-US

**Repository** CRAN

**Date/Publication** 2024-06-24 09:00:02 UTC

## Contents

GeomArchetypal-package . . . . .	2
BLB_archetypal . . . . .	4
closer_grid_archetypal . . . . .	6
fast_archetypal . . . . .	8
gallupGPS6 . . . . .	9
grid_archetypal . . . . .	10
plot.closer_grid_archetypal . . . . .	12
plot.grid_archetypal . . . . .	13
points_inside_convex_hull . . . . .	14
print.BLB_archetypal . . . . .	15
print.closer_grid_archetypal . . . . .	16
print.grid_archetypal . . . . .	17
summary.BLB_archetypal . . . . .	18
summary.closer_grid_archetypal . . . . .	19
summary.grid_archetypal . . . . .	20
<b>Index</b>	<b>21</b>

---

GeomArchetypal-package

*Finds the Geometrical Archetypal Analysis of a Data Frame*

---

### Description

Performs Geometrical Archetypal Analysis after creating Grid Archetypes which are the Cartesian Product of all minimum, maximum variable values. Since the archetypes are fixed now, we have the ability to compute the convex composition coefficients for all our available data points much faster by using the half part of PCHA method. Additionally we can decide to keep as archetypes the closer to the Grid Archetypes ones. Finally the number of archetypes is always 2 to the power of the dimension of our data points if we consider them as a vector space.

### Details

Given a data frame `df` which is a matrix of  $n$  observations (rows) for the  $d$  variables (columns) we compute for all variables  $X_j$  the  $(X_{j.min}, X_{j.max})$ ,  $j=1,2,\dots, n$ .

By taking the Cartesian Product of all those sets we form the vector set of Grid Archetypes which are 2 to the power of  $d$  and all other points lie inside their Convex Hull.

For example if we take the case of  $d=2$  and our variables are named  $X, Y$ , then the Cartesian Product gives next points:

$(X_{min}, Y_{min}), (X_{max}, Y_{min}), (X_{min}, Y_{max}), (X_{max}, Y_{max})$

Now the problem of seeking for the best number of archetypes is solved and  $k$  is 2 to the power of  $d$ .



```

# Print
print(cga)
# Summary
summary(cga)
# Plot
plot(cga)
# Fast Archetypal:
# we use as archetypal rows the closer to the Grid Archetypes
# as they were found by closer_grid_archetypal() function
fa=fast_archetypal(df, irows = cga$grid_rows, diag_less = 1e-3,
                  niter = 200, use_seed = vseed)

# Print
print(fa)
# Summary
summary(fa)
# Plot
plot(fa)

```

---

BLB\_archetypal

*Archetypal Analysis using the Bag of Little Bootstraps*


---

## Description

Archetypal analysis using the bag of little bootstraps as the resampling approach, following [1]

## Usage

```

BLB_archetypal(df = NULL, group_var = NULL,
aa_var = NULL, use_seed = NULL,
b = 0.6, n = 20, r = 100, n_core = 1,
n_iter = 30, ci_sigma = 2,
n_tails = 10, max_cor = 0.3,
verbose = TRUE, diag_less = 1e-2)

```

## Arguments

df	The data frame with the original sample to be processed
group_var	Draw the subsample equally from groups (integer or character)
aa_var	Character vector of the variable names that will be used
use_seed	Integer, if not NULL, used as set.seed() for reproducibility
b	Numeric, set size of subsample i.e. = nrow(df)^b (default 0.6)
n	Integer, number of subsamples to generate (default 20)
r	Integer, number of bootstraps of each subsample (default 100)
n_core	Integer, number of cores used for archetypal analysis of bootstraps
n_iter	Integer, number of iterations for fast_archetypal
ci_sigma	Integer, for empirical confidence intervals

n_tails	Integer, minimum number of bootstrap estimates required in tails for robust interval estimates (default 10 each tail)
max_cor	Default = 0.20, numeric for warning on orthogonality
verbose	Logical, reports progress of each subsample and batch of bootstraps
diag_less	The expected mean distance from 1 for the diagonal elements of submatrix $A[1:kappas,:]$

### Details

Note 1.

Without the weighted analysis idea of Kleiner et al which is inappropriate for geometrically-based archetypal analysis

Note 2.

The archetypes are defined from the minimums and maximums of the data to provide a fixed frame of reference for resampling. Resampling variation is thus simplified and only concerns compositions.

Note 3.

Assumes grouped data but allows user to supply a group variable with only one value

### Value

An object of class "BLB\_archetypal" which is a list with next members:

1. arches, the Grid Archetypes
2. aa\_tests, the run statistics for all subsamples, batches and replications (bootstraps)
3. pop\_compos, the population estimates of compositions (by group or without grouping)
4. lower\_ci, the lower confidence interval at the ci\_sigma sigma level
5. upper\_ci, the upper confidence interval at the ci\_sigma sigma level
6. ci\_sigma, the ci\_sigma level for confidence intervals

### Author(s)

David. F. Midgley

### References

[1] Ariel Kleiner, Ameet Talwalkar, Purnamrita Sarkar, Michael I. Jordan, [doi:10.1111/rssb.12050](https://doi.org/10.1111/rssb.12050)

### See Also

[closer\\_grid\\_archetypal](#), [grid\\_archetypal](#), [fast\\_archetypal](#)

**Examples**

```

{
# Load package
library(GeomArchetypal)
# Load data
data("gallupGPS6")
# draw a small sample
set.seed(2024)
df <- GallupGPS6[sample(1:nrow(gallupGPS6),35000,replace = FALSE),]
# invent a grouping variable
df$grp <- cut(df$risktaking, breaks = 2)
test <- BLB_archetypal(df = df,
                      group_var = "grp",
                      aa_var = c("patience","risktaking","trust"),
                      n = 1, r = 2, n_core = 1,
                      diag_less = 1e-2)
# will generate a warning because number of bootstraps is too small to
# estimate default confidence intervals
# Print results of the "BLB_archetypal" class object:
print(test)
# Summarize the "BLB_archetypal" class object:
summary(test)
}

```

---

closer\_grid\_archetypal

*Performs the Archetypal Analysis of a Data Frame by using as Archetypes the Closer to The Archetypal Grid Data Points*

---

**Description**

the closer points to the archetypal grid are used as archetypes and then every data point is being expressed as a convex combination of those by using a modified PCHA method.

**Usage**

```

closer_grid_archetypal(dg,
                      diag_less = 1e-2,
                      niter=30,
                      use_seed = NULL,
                      verbose = TRUE)

```

**Arguments**

dg	The data frame with dimensions n x d
diag_less	The expected mean distance from 1 for the diagonal elements of submatrix A[irows,:], where irows are the closer to the Grid Archetypal data rows.

niter	The number of iterations that the A-update step should be done.
use_seed	If it is not NULL, then is used at the set.seed() for reproducibility reasons
verbose	If it is set to TRUE, then both initialization and iteration details are printed out

### Details

The archetypal grid is being computed by taking the expand grid of the  $[X_{imin}, X_{imax}]$ ,  $i=1, \dots, d$  of all available variables. Then distances of all data points from that grid are calculated and the closer set of vectors is chosen.

### Value

An object of class `closer_grid_archetypal` which is a list with members:

1. `grid`, the archetypal grid
2. `grid_rows`, the rows of the data frame that formed the archetypal grid
3. `aa`, an object of class `archetypal`

### See Also

[grid\\_archetypal](#)

### Examples

```
# Load package
library(GeomArchetypal)
# Create random data
set.seed(20140519)
df=matrix(runif(90) , nrow = 30, ncol=3)
colnames(df)=c("x", "y", "z")
# Closer Grid Archetypal
cga=closer_grid_archetypal(df,
  diag_less = 1e-2,
  niter = 150,
  verbose = FALSE)
# Print the class "closer_grid_archetypal":
print(cga)
# Summary of the class "closer_grid_archetypal":
summary(cga)
# Plot the class "closer_grid_archetypal":
plot(cga)
# Observe the Closer Grid Archetypes near the 8 corners of the cube ...
```

---

fast_archetypal	<i>Performs the Archetypal Analysis of a Data Frame by using a Given Set of Archetypes</i>
-----------------	--

---

### Description

Performs the archetypal analysis of a data frame by using a known set of archetypes as rows of the data matrix.

### Usage

```
fast_archetypal(df,  
                irows,  
                diag_less = 1e-2,  
                niter = 30,  
                verbose = TRUE,  
                data_tables = TRUE,  
                use_seed = NULL)
```

### Arguments

df	The data frame with dimensions $n \times d$
irows	The rows from data frame that represent the archetypes
diag_less	The expected mean distance from 1 for the diagonal elements of submatrix $A[irows, :]$
niter	The number of times that the A-update process should be done
verbose	If it is set to TRUE, then both initialization and iteration details are printed out
data_tables	If it set to TRUE, then a data table for the initial data points will be computed
use_seed	If it is not NULL, then is used at the <code>set.seed()</code> for reproducibility reasons

### Details

If we know the archetypes, then we can bypass the half part of PCHA and perform only the A-update part, that of computing the convex combinations for each data point. Then archetypal analysis is a fast procedure, since we need only to compute one matrix.

### Value

An object of class 'archetypal' is returned.

### See Also

[grid\\_archetypal](#), [closer\\_grid\\_archetypal](#)

**Examples**

```
# Load package
library(GeomArchetypal)
# Create random data
set.seed(20140519)
df=matrix(runif(90) , nrow = 30, ncol=3)
colnames(df)=c("x","y","z")
# Closer Grid Archetypal
cga=closer_grid_archetypal(df, diag_less = 1e-3,
                           niter = 250, verbose = FALSE)
# The closer to the Grid Archetypes points - rows are:
crows = cga$grid_rows
print(crows)
# Now we call the fast_archetypal() with those rows as argument:
fa=fast_archetypal(df, irows = crows, diag_less = 1e-3,
                  niter = 250, verbose = FALSE)

# Print:
print(fa)
# Summary:
summary(fa)
# Plot:
plot(fa)
# Results are identical to the closer_grid_archetypal() ones:
all.equal(cga$aa$BY,fa$BY)
```

---

gallupGPS6

*Gallup Global Preferences Study processed data set of six variables*

---

**Description**

A 76132 x 6 data frame derived from Gallup Global Preferences Study, see [1] and [2] for details. It can be used as a big data set example.

**Usage**

```
data("gallupGPS6")
```

**Format**

A data frame with 76132 complete observations on the following 6 variables.

patience a numeric vector  
risktaking a numeric vector  
posrecip a numeric vector  
negrecip a numeric vector  
altruism a numeric vector  
trust a numeric vector

**Details**

Data processing:

1. The non complete rows have been removed
2. The duplicated rows have also been removed

**Note**

1. The data was provided under a Creative Commons NonCommerical ShareAlike 4.0 license: <https://creativecommons.org/licenses/by-nc-sa/4.0/>
2. Other variables and identifiers from the original data have been dropped

**Source**

Individual data set was downloaded from

<https://www.gallup.com/analytics/318923/world-poll-public-datasets.aspx>, last accessed 2024-03-09.

**References**

[1] Falk, A., Becker, A., Dohmen, T., Enke, B., Huffman, D., & Sunde, U. (2018). Global evidence on economic preferences. *Quarterly Journal of Economics*, 133 (4), 1645-1692.

[2] Falk, A., Becker, A., Dohmen, T. J., Huffman, D., & Sunde, U. (2016). The preference survey module: A validated instrument for measuring risk, time, and social preferences. IZA Discussion Paper No. 9674.

**Examples**

```
# Load package
library(GeomArchetypal)
data(gallupGPS6)
summary(gallupGPS6)
```

---

grid_archetypal	<i>Performs the Archetypal Analysis of a Data Frame by using as Archetypes the Archetypal Grid</i>
-----------------	--

---

**Description**

The archetypal grid is the expand grid of all intervals  $(X.i_{min}, X.i_{max})$ ,  $i=1, \dots, d$  for a  $d$ -dimensional data frame.

That grid is used as archetypes and then only the A-update part of PCHA algorithm is used for computing the compositions of all data points.

The number of archetypes is always  $kappa = 2^d$ .

**Usage**

```
grid_archetypal(dg,
                diag_less = 1e-2,
                niter = 30,
                use_seed = NULL,
                verbose = TRUE)
```

**Arguments**

dg	The data frame with dimensions n x d
diag_less	The expected mean distance from 1 for the diagonal elements of submatrix A[1:kappas,;]
niter	The number of times that the A-update process should be done
use_seed	If it is not NULL, then is used at the set.seed() for reproducibility reasons
verbose	If it is set to TRUE, then both initialization and iteration details are printed out

**Details**

The archetypal grid defines a hyper-volume which contains the 100 % of all data points, if we take those grid points as the Convex Hull of all points. Although the archetypal grid points do not necessarily lie inside the data frame, here we do not care about that property: we only seek for the matrix of convex combinations (or composition matrix) A.

**Value**

An object of class grid\_archetypal which is a list with members:

1. grid, the archetypal grid
2. aa, an object of class 'archetypal' which includes the archetypal grid as the first 2^d rows
3. A, the A-matrix with dimensions n x d that defines the compositions of all data points
4. Y, the matrix of initial data points

**See Also**

[closer\\_grid\\_archetypal](#)

**Examples**

```
# Load package
library(GeomArchetypal)
# Create random data:
set.seed(20140519)
df=matrix(runif(90) , nrow = 30, ncol=3)
colnames(df)=c("x","y","z")
# Grid Archetypal:
gaa=grid_archetypal(df, diag_less = 1e-6,
                    niter = 70, verbose = FALSE)
# Print class "grid_archetypal":
```

```

gaa
# Summary class "grid_archetypal":
summary(gaa)
# Plot class "grid_archetypal":
plot(gaa)
# Observe the Grid Archetypes at the 8 corners of the cube ..

```

---

```
plot.closer_grid_archetypal
```

*Plot an Object of the Class closer\_grid\_archetypal*

---

## Description

It plots the output of [closer\\_grid\\_archetypal](#)

## Usage

```
## S3 method for class 'closer_grid_archetypal'
plot(x, ...)
```

## Arguments

x	An object of the class closer_grid_archetypal
...	Other arguments (ignored)

## Details

Given an object of class closer\_grid\_archetypal the archetypal analysis result is plotted.

## Value

No return value, called for side effects

## Examples

```

# Load package
library(GeomArchetypal)
# Create random data
set.seed(20140519)
df=matrix(runif(90) , nrow = 30, ncol=3)
colnames(df)=c("x","y","z")
# Closer Grid Archetypal
cga=closer_grid_archetypal(df, niter = 70, verbose = FALSE)
# Plot the class "closer_grid_archetypal":
plot(cga)

```

---

plot.grid\_archetypal *Plot an Object of the class grid\_archetypal*

---

### Description

It plots the output of [grid\\_archetypal](#)

### Usage

```
## S3 method for class 'grid_archetypal'  
plot(x, ...)
```

### Arguments

x	An object of the class grid_archetypal
...	Other arguments (ignored)

### Details

Given an object of class grid\_archetypal the archetypal analysis result is plotted.  
Remark: the first  $2^d$  rows of the input data frame has Grid Archetypes (d is the dimension of the data points).

### Value

No return value, called for side effects

### Examples

```
# Load package  
library(GeomArchetypal)  
# Create random data:  
set.seed(20140519)  
df=matrix(runif(90) , nrow = 30, ncol=3)  
colnames(df)=c("x","y","z")  
# Grid Archetypal:  
gaa=grid_archetypal(df, niter = 70, verbose = FALSE)  
# Plot the class "archetypal":  
plot(gaa)
```

---

`points_inside_convex_hull`

*Computes the Percentage of Points that Lie Inside the Convex Hull which is Created by a Set of Vectors*

---

### Description

Given a set of  $k$   $d$ -dimensional vectors which creates a Convex Hull (CH) we want to find the percentage of the  $n$  points of the  $n \times d$  data frame `df` that lie inside that CH.

### Usage

```
points_inside_convex_hull(df, dp)
```

### Arguments

<code>df</code>	The $n \times d$ data frame of all available data points
<code>dp</code>	The $k \times d$ data frame of the given set of points that creates the Convex Hull

### Details

In order for a really Convex Hull creation it must hold that:  $k \geq d + 1$ , otherwise the problem is not well stated.

### Value

A numeric output with percentage in two decimal digits

### Note

Keep in mind that working with dimension greater than 6 will practical lead to extreme time executions. It highly suggested to work only for spaces with  $d \leq 6$ .

### Author(s)

Demetris T. Christopoulos

### Examples

```
# Load package
library(GeomArchetypal)
# Create random data:
set.seed(20140519)
df=matrix(runif(90) , nrow = 30, ncol=3)
colnames(df)=c("x","y","z")
# Grid Archetypal:
gaa=grid_archetypal(df, niter = 70, verbose = FALSE)
pc1=points_inside_convex_hull(df, gaa$grid)
```

```

print(pc1)
# [1] 100
# Closer Grid Archetypal:
cga=closer_grid_archetypal(df, niter = 70, verbose = FALSE)
pc2=points_inside_convex_hull(df, cga$aa$BY)
print(pc2)
# [1] 59

```

---

```
print.BLB_archetypal Print an Object of the Class BLB_archetypal
```

---

## Description

It prints the output of [BLB\\_archetypal](#)

## Usage

```
## S3 method for class 'BLB_archetypal'
print(x, ...)
```

## Arguments

x	An object of the class BLB_archetypal
...	Other arguments (ignored)

## Details

Given an object of class BLB\_archetypal all the results are printed in explanatory form.

## Value

No return value, called for side effects

## Examples

```

{
# Load package
library(GeomArchetypal)
# Load data
data("gallupGPS6")
# draw a small sample
set.seed(2024)
df <- GallupGPS6[sample(1:nrow(GallupGPS6), 35000, replace = FALSE),]
# invent a grouping variable
df$grp <- cut(df$risktaking, breaks = 2)
test <- BLB_archetypal(df = df,
                      group_var = "grp",
                      aa_var = c("patience", "risktaking", "trust"),
                      n = 1, r = 2, n_core = 1,

```

```

        diag_less = 1e-2)
# Print the results of the class "BLB_archetypal" object:
print(test)

}

```

---

```
print.closer_grid_archetypal
```

*Print an Object of the closer\_grid\_archetypal*

---

## Description

It prints the output of [closer\\_grid\\_archetypal](#)

## Usage

```
## S3 method for class 'closer_grid_archetypal'
print(x, ...)
```

## Arguments

x	An object of the class closer_grid_archetypal
...	Other arguments (ignored)

## Details

Given an object of class closer\_grid\_archetypal all the results are printed in explanatory form.

## Value

No return value, called for side effects

## Examples

```

# Load package
library(GeomArchetypal)
# Create random data
set.seed(20140519)
df=matrix(runif(90) , nrow = 30, ncol=3)
colnames(df)=c("x","y","z")
# Closer Grid Archetypal
cga=closer_grid_archetypal(df, niter = 70, verbose = FALSE)
# Print the class "closer_grid_archetypal"
print(cga)

```

---

print.grid\_archetypal *Print an Object of the Class grid\_archetypal*

---

## Description

It prints the output of [grid\\_archetypal](#)

## Usage

```
## S3 method for class 'grid_archetypal'  
print(x, ...)
```

## Arguments

x	An object of the class grid_archetypal
...	Other arguments (ignored)

## Details

Given an object of class grid\_archetypal all the results are printed in explanatory form.

## Value

No return value, called for side effects

## Examples

```
# Load package  
library(GeomArchetypal)  
# Create random data:  
set.seed(20140519)  
df=matrix(runif(90) , nrow = 30, ncol=3)  
colnames(df)=c("x", "y", "z")  
# Grid Archetypal:  
gaa=grid_archetypal(df, niter = 70, verbose = FALSE)  
# Print the class "grid_archetypal":  
print(gaa)
```

---

`summary.BLB_archetypal`*Summarize an Object of the Class BLB\_archetypal*

---

## Description

It summarizes the output of [BLB\\_archetypal](#)

## Usage

```
## S3 method for class 'BLB_archetypal'  
summary(object, ...)
```

## Arguments

<code>object</code>	An object of the class <code>BLB_archetypal</code>
<code>...</code>	Other arguments (ignored)

## Details

Given an object of class `BLB_archetypal` all the results are being summarized in explanatory form.

## Value

No return value, called for side effects

## Examples

```
{  
# Load package  
library(GeomArchetypal)  
# Load data  
data("gallupGPS6")  
# draw a small sample  
set.seed(2024)  
df <- GallupGPS6[sample(1:nrow(GallupGPS6),35000,replace = FALSE),]  
# invent a grouping variable  
df$grp <- cut(df$risktaking, breaks = 2)  
test <- BLB_archetypal(df = df,  
  group_var = "grp",  
  aa_var = c("patience","risktaking","trust"),  
  n = 1, r = 2, n_core = 1,  
  diag_less = 1e-2)  
# Summarize the results of the "BLB_archetypal" class object:  
summary(test)  
}
```

---

`summary.closer_grid_archetypal`*Summary of an Object of the Class closer\_grid\_archetypal*

---

## Description

It gives a summary for the output of `closer_grid_archetypal`

## Usage

```
## S3 method for class 'closer_grid_archetypal'  
summary(object, ...)
```

## Arguments

<code>object</code>	An object of the class <code>closer_grid_archetypal</code>
<code>...</code>	Other arguments (ignored)

## Details

Given an object of class `closer_grid_archetypal` the summary of the archetypal analysis output is given.

## Value

No return value, called for side effects

## Examples

```
# Load package  
library(GeomArchetypal)  
# Create random data  
set.seed(20140519)  
df=matrix(runif(90) , nrow = 30, ncol=3)  
colnames(df)=c("x","y","z")  
# Closer Grid Archetypal  
cga=closer_grid_archetypal(df, niter = 70, verbose = FALSE)  
# Summary of the class "closer_grid_archetypal":  
summary(cga)
```

---

`summary.grid_archetypal`*Summary of an Object of the Class grid\_archetypal*

---

## Description

It gives a summary for the output of `grid_archetypal`

## Usage

```
## S3 method for class 'grid_archetypal'  
summary(object, ...)
```

## Arguments

<code>object</code>	An object of the class <code>grid_archetypal</code>
<code>...</code>	Other arguments (ignored)

## Details

Given an object of class `grid_archetypal` the summary of the archetypal analysis output is given.  
Remark: the first  $2^d$  rows of the input data frame are the Grid Archetypes ( $d$  is the dimension of the data points).

## Value

No return value, called for side effects

## Examples

```
# Load package  
library(GeomArchetypal)  
# Create random data:  
set.seed(20140519)  
df=matrix(runif(90) , nrow = 30, ncol=3)  
colnames(df)=c("x","y","z")  
# Grid Archetypal:  
gaa=grid_archetypal(df, niter = 70, verbose = FALSE)  
# Summary of the class "grid_archetypal":  
summary(gaa)
```

# Index

- \* **datasets**
  - gallupGPS6, [9](#)
- \* **grid**
  - GeomArchetypal-package, [2](#)
- BLB\_archetypal, [4](#), [15](#), [18](#)
- closer\_grid\_archetypal, [3](#), [5](#), [6](#), [8](#), [11](#), [12](#),  
[16](#), [19](#)
- fast\_archetypal, [3](#), [5](#), [8](#)
- gallupGPS6, [9](#)
- GeomArchetypal
  - (GeomArchetypal-package), [2](#)
- GeomArchetypal-package, [2](#)
- grid\_archetypal, [3](#), [5](#), [7](#), [8](#), [10](#), [13](#), [17](#), [20](#)
- plot.closer\_grid\_archetypal, [12](#)
- plot.grid\_archetypal, [13](#)
- points\_inside\_convex\_hull, [14](#)
- print.BLB\_archetypal, [15](#)
- print.closer\_grid\_archetypal, [16](#)
- print.grid\_archetypal, [17](#)
- summary.BLB\_archetypal, [18](#)
- summary.closer\_grid\_archetypal, [19](#)
- summary.grid\_archetypal, [20](#)