

# Package: image2data (via r-universe)

November 5, 2024

**Type** Package

**Title** Turn Images into Data Sets

**Version** 1.0.1

**Description** The goal of 'image2data' is to extract images and return them into a data set, especially for teaching data manipulation and data visualization. Basically, the eponymous function takes an image file ('png', 'tiff', 'jpeg', 'bmp') and turn it into a data set, pixels being rows (subjects) and columns (variables) being their coordinate positions (x- and y-axis) and their respective color (in hex codes). The function can return a complete image or a range of color (i.e., contour, silhouette). The data can then be manipulated as would any data set by either creating other related variables (to hide the image) or as a genuine toy data set.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.2

**Imports** readbitmap (>= 0.1.0)

**Config/pak/sysreqs** libjpeg-dev libpng-dev libtiff-dev

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

**RemoteUrl** <https://github.com/quantmeth/image2data>

**RemoteRef** HEAD

**RemoteSha** 50c8a875a08f57dc1fe8febb351fcf24fbf6786

## Contents

image2data	2
Index	4

---

 image2data

*Turn an image into data*


---

### Description

Extract an image file ("png", "tiff", "jpeg", "bmp") and turn it into an enjoyable data set, pixels being rows (subjects) and columns (variables) being their coordinate positions (x and y axis) and their respective color (in hex codes).

### Usage

```
image2data(
  path,
  type = "fill",
  scaling = "standardized",
  showplot = TRUE,
  reduce = 1,
  A = 1,
  R = c(0, 0.05),
  G = c(0, 0.05),
  B = c(0, 0.05),
  Grey = NULL,
  precision = 1,
  seed = NULL
)
```

### Arguments

path	Path to image file.
type	Type of extraction of data. type = "fill" (default) returns the complete image as data whereas type = "line" returns a specific range of color (default is black).
scaling	Transform the data to a specified scale. Three options are available: "standardized", "original", "normalized". scaling = "standardized" converts data in a standardized form, $\mu = 0, \sigma = 1$ (default); scaling = "normalized" converts data in a normalized form (to unit vectors); and scaling = "original" keeps the data untransformed.
showplot	Show a preliminary plot of the data (default is TRUE).
reduce	reduce can be a number reduce > 0 or reduce = "unique". By default reduce = 1, so all pixels are returned. Specified values between 0 to 1 will return the corresponding proportion of the pixels. Values over 1 will return the number of pixels (e.g., reduce = 3 returns 3 data). If the chosen number is over the number of pixels, then random duplicates are added. If reduce = "unique" only unique elements (given a certain precision) are returned.

A	Transparency, otherwise known as $\alpha$ . By default, only non transparent ( $A = 1$ ) values are returned. Semi-transparent colors ( $0 < A < 1$ ) are supported. Values between the A to 1 range will be return. If $A = 0$ , all pixels are returned regardless of transparency.
R, G, B	Color to return with <code>type = "line"</code> (the default range is <code>c(0, .05)</code> for each, i.e., black). A single "range" of color can be used.
Grey	Grey range to be returned with <code>type = "line"</code> . Grey overwrites R, G, B and behaves similarly. Default is NULL
precision	Set precision of <code>reduce = "unique"</code> . Default is 1. It can be any integer $>0$ . Values closer to zero are less precised (less data), higher values are more precise (more data).
seed	Set seed value for random pixel returned with <code>reduce</code> .

### Value

A data frame with pixels as rows and columns are x and y coordinates and g is their color in hex (factors).

### Examples

```
path <- system.file(file.path("extdata", "success.png"), package = "image2data")
image2data(path = path, type = "line")
image2data(path = path, type = "line", Grey = c(0,.50))

## Not run:
image2data(path = file.choose())

## End(Not run)
```

# Index

image2data, [2](#)