

Package: kraw (via r-universe)

October 5, 2026

Title Explore CR2 Raw Images

Version 0.9.0

Description Read, extract & visualize metadata information from CR2 raw camera images.

License GPL-3 + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.1.0)

Imports dplyr, ggplot2, ggforce, ggpubr, digest, see

Config/roxygen2/version 8.1.0

URL <https://thekangaroofactory.github.io/kraw/>

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev make libicu-dev

Repository <https://thekangaroofactory.r-universe.dev>

Date/Publication 2026-10-05 15:47:30 UTC

RemoteUrl <https://github.com/thekangaroofactory/kraw>

RemoteRef HEAD

RemoteSha 48554dee35d4767fbbc3d507c2ba001ea9f37e40

Contents

header	2
ifd	3
ifd_entry	3
most_frequent	4
order_bytes	5
p_aperture	6
p_camera	7
p_focal	8
p_iso	9
p_lens	10

p_orientation 10

p_overview 11

p_shutter_speed 12

p_theme 13

raw_bytes 13

read_cr2 14

scan 15

tag_id 15

to_num 16

Index 17

header	<i>CR2 Headers</i>
--------	--------------------

Description

Extract the TIFF and Canon headers.

Usage

header(x)

Arguments

x a raw vector.

Details

The function returns a list with the following elements:

- endianness: order in which the bites are written (expecting little endian / II)
- magic_number: TIFF signature (expecting 42)
- offset_first_ifd: offset to first ifd
- cr_marker: raw marker (expecting 'CR+2')
- cr_version: raw marker version
- offset_ifd_raw: offset to first raw ifd

Value

a list.

Examples

```
## Not run:  
header(x)  
  
## End(Not run)
```

ifd	<i>Image File Directory</i>
-----	-----------------------------

Description

Image File Directory

Usage

```
ifd(x, offset = 1)
```

Arguments

x	a raw vector.
offset	an integer to indicate where the sequence starts.

Details

The output list contains:

- nb_entries: the number of entries
- entries: the entries
- offset_next_ifd: offset to the next ifd.

Value

a list with the IFD information.

Examples

```
## Not run:  
ifd(x, offset = 924)  
  
## End(Not run)
```

ifd_entry	<i>IFD Entry Value or Offset</i>
-----------	----------------------------------

Description

IFD Entry Value or Offset

Usage

```
ifd_entry(ifd, id, raw_vector = NULL, verbose = FALSE)
```

Arguments

ifd	an IFD list.
id	the id of the entry.
raw_vector	the raw vector to extract value when an offset is involved.
verbose	whether to turn ON/OFF console log.

Details

ifd expects a list as returned by the `ifd()` function.

Value

a numeric value or a vector with the offset.

See Also

`ifd()`

Examples

```
## Not run:
ifd_entry(ifd, id = "0100")

## End(Not run)
```

most_frequent	<i>Most Frequent</i>
---------------	----------------------

Description

Extract most frequent combination.

Usage

```
most_frequent(data)
```

Arguments

data	a data.frame (output of <code>scan()</code>).
------	--

Details

The function computes the most frequent combination over: camera, orientation, exposure_time, f_number, iso_speed, lens_model, focal_length

Value

a data.frame.

Examples

```
## Not run:
most_frequent(scan("."))

## End(Not run)
```

order_bytes	<i>Order Bytes</i>
-------------	--------------------

Description

Order Bytes

Usage

```
order_bytes(x, endian = "II")
```

Arguments

x	a vector or data.frame of bytes.
endian	the endianness.

Details

When x is a data.frame, value will be a data.frame, each row being the concatenation of the ordered columns.

Only little endian (i.e. "II") is supported.

Value

the corresponding hexadecimal vector or data.frame.

Examples

```
## Not run:
order_bytes(x, endian = "II")

## End(Not run)
```

p_aperture	<i>Aperture Plot</i>
------------	----------------------

Description

Aperture Plot

Usage

```
p_aperture(data, fg = "grey", bg = "grey", theme = p_theme())
```

Arguments

data	a data.frame (see details).
fg	a foreground color.
bg	a background color.
theme	an optional theme function.

Details

data expects the following columns:

- lens_model: character, the name of the lens
- f_number: character, the aperture value

Value

a 'ggplot' object.

Examples

```
## Not run:  
p_aperture(data)  
  
## End(Not run)
```

p_camera

Camera Plot

Description

Camera Plot

Usage

```
p_camera(data, fg = NA, bg = "grey", theme = p_theme())
```

Arguments

data	a data.frame (see details).
fg	a foreground color.
bg	a background color.
theme	an optional theme function.

Details

data expects a data.frame with the following columns:

- camera: character, the camera name
- n: numeric, the number of images for the camera

Value

a 'ggplot' object.

See Also

[p_theme\(\)](#)

Examples

```
p_camera(data.frame(camera = "Canon EOS 70D", n = 10))
```

p_focal	<i>Focal Length Plot</i>
---------	--------------------------

Description

Focal Length Plot

Usage

```
p_focal(data, fg = "grey", bg = "grey", theme = p_theme())
```

Arguments

data	a data.frame (see details).
fg	a foreground color.
bg	a background color.
theme	an optional theme function.

Details

data expects the following columns:

- lens_model: character, the name of the lens
- focal_length: numeric, the focal length
- n: the number of images.

Value

a 'ggplot' object.

Examples

```
## Not run:  
p_focal(data = metadata |>  
  dplyr::group_by(lens_model, focal_length) |>  
  dplyr::summarise(n = dplyr::n()))  
  
## End(Not run)
```

p_iso	<i>ISO Plot</i>
-------	-----------------

Description

ISO Plot

Usage

```
p_iso(data, bg = "grey", theme = p_theme())
```

Arguments

data	a data.frame (see details).
bg	a background color.
theme	an optional theme function.

Details

data expects the following columns:

- lens_model: character, the name of the lens
- iso_speed: numeric, the ISO speed
- n: the number of images.

Value

a 'ggplot' object.

Examples

```
## Not run:  
p_iso(data.frame(lens_model = "foo", iso_speed = 100, n = 3))  
  
## End(Not run)
```

p_lens	<i>Lens Plot</i>
--------	------------------

Description

Raw images per lens type.

Usage

```
p_lens(data, fg = "#000", bg = "grey", theme = p_theme())
```

Arguments

data	a data.frame (see details).
fg	a foreground color.
bg	a background color.
theme	an optional theme function.

Details

data expects the following columns:

- lens_model: a character string, the name of the lens
- n: numeric, the number of images for this lens

Value

a 'ggplot' object.

See Also

[p_theme\(\)](#)

p_orientation	<i>Orientation Plot</i>
---------------	-------------------------

Description

Distribution over landscape and portrait.

Usage

```
p_orientation(nl, np, fg = "black", bg = "grey", theme = p_theme())
```

Arguments

nl	a numeric, the number of landscape.
np	a numeric, the number of portraits.
fg	a foreground color.
bg	a background color.
theme	an optional theme function.

Value

a 'ggplot' object.

See Also

[p_theme\(\)](#)

Examples

```
p_orientation(nl = 42, np = 112)
```

p_overview

Overview Plot

Description

Overview Plot

Usage

```
p_overview(metadata)
```

Arguments

metadata	a data.frame (see details).
----------	-----------------------------

Details

metadata expects an output from the [scan\(\)](#) function.

Value

a ggplot object.

See Also

[scan\(\)](#)

Examples

```
## Not run:  
viz(scan(path = "."))  
  
## End(Not run)
```

p_shutter_speed	<i>Shutter Speed Plot</i>
-----------------	---------------------------

Description

Shutter Speed Plot

Usage

```
p_shutter_speed(data, fg = "grey", bg = "grey", theme = p_theme())
```

Arguments

data	a data.frame (see details).
fg	a foreground color.
bg	a background color.
theme	an optional theme function.

Details

data expects the following columns:

- lens_model: character, the name of the lens
- exposure_time: character, the aperture value
- n: the number of images.

Value

a 'ggplot' object.

Examples

```
## Not run:  
p_shutter_speed(data)  
  
## End(Not run)
```

p_theme

Default Theme

Description

Default Theme

Usage

```
p_theme()
```

Details

The function is used as a default theme for the plots, which can be replaced by a custom one.

Value

a 'ggplot' theme.

Examples

```
p_theme()
```

raw_bytes

Sequence of Bytes

Description

Sequence of Bytes

Usage

```
raw_bytes(x, offset = 0, n = 1)
```

Arguments

x	a vector of raw bytes.
offset	an integer, to indicate the lower limit of the sequence.
n	an integer, to indicate the length of the sequence.

Details

Note that offset = 0 (default) is the index of the first element of the list.

Value

a vector of raw bytes.

Examples

```
## Not run:
# get a sequence of 4 bytes at offset 294
raw_bytes(x, offset = 294, n = 4)

## End(Not run)
```

read_cr2	<i>Read CR2 File</i>
----------	----------------------

Description

Read CR2 File

Usage

```
read_cr2(
  file,
  image = FALSE,
  mapping_exif = mapping_exif,
  mapping_canon = mapping_canon
)
```

Arguments

- file the file to read (including its path).
- image a logical whether the images should be loaded or not (default FALSE).
- mapping_exif a data.frame of the EXIF tag mapping.
- mapping_canon a data.frame of the Canon tag mapping.

Value

a list.

Examples

```
## Not run:
read_cr2(file = "C:/download/example.CR2")

## End(Not run)
```

scan	<i>Scan Folder</i>
------	--------------------

Description

Scan Folder

Usage

scan(path)

Arguments

path the path to scan.

Value

a data.frame of the metadata.

Examples

```
## Not run:
report <- scan(path = ".")

## End(Not run)
```

tag_id	<i>Tag Id</i>
--------	---------------

Description

Tag Id

Usage

tag_id(mapping, x)

Arguments

mapping a data.frame for the mapping.
x a character string, the name of the key.

Value

a character string.

Examples

```
## Not run:  
tag_id(mapping_exif, "ImageWidth")  
  
## End(Not run)
```

to_num	<i>Convert</i>
--------	----------------

Description

Convert

Usage

```
to_num(x)
```

Arguments

x a character vector.

Value

a vector.

Examples

```
to_num("ffff")
```

Index

header, [2](#)

ifd, [3](#)

ifd(), [4](#)

ifd_entry, [3](#)

most_frequent, [4](#)

order_bytes, [5](#)

p_aperture, [6](#)

p_camera, [7](#)

p_focal, [8](#)

p_iso, [9](#)

p_lens, [10](#)

p_orientation, [10](#)

p_overview, [11](#)

p_shutter_speed, [12](#)

p_theme, [13](#)

p_theme(), [7](#), [10](#), [11](#)

raw_bytes, [13](#)

read_cr2, [14](#)

scan, [15](#)

scan(), [4](#), [11](#)

tag_id, [15](#)

to_num, [16](#)