Package ‘metaDigitise’

July 29, 2018

Title Extract and Summarise Data from Published Figures
Version 1.0.0
Description High-throughput, flexible and reproducible extraction of data from figures in primary research papers. metaDigitise() can extract data and / or automatically calculate summary statistics for users from box plots, bar plots (e.g., mean and errors), scatter plots and histograms.
Depends R (>= 3.4)
License GPL (>= 2)
Encoding UTF-8
LazyData true
RoxygenNote 6.0.1
Imports magick, stats, graphics, utils, purrr
Suggests mockery, testthat, knitr, rmarkdown
BugReports https://github.com/daniel1noble/metaDigitise/issues
VignetteBuilder knitr
NeedsCompilation no
Author Joel Pick [aut],
      Shinichi Nakagawa [aut],
      Daniel Noble [aut, cre]
Maintainer Daniel Noble <daniel.wa.noble@gmail.com>
Repository CRAN
Date/Publication 2018-07-29 08:40:03 UTC

R topics documented:

ask_variable .......................................................... 3
bulk_edit ............................................................. 3
calibrate ............................................................... 4
cal_coords ............................................................ 4
cat_matrix ............................................................ 5
CI95_to_sd ............................................................. 5
R topics documented:

- `convert_group_data` ................................................................. 6
- `convert_histogram_data` ........................................................ 6
- `delete_group` ......................................................................... 6
- `dir_details` ............................................................................. 7
- `edit_group` ............................................................................... 8
- `edit_metaDigitise` .................................................................. 8
- `enter_N` ................................................................................... 9
- `error_to_sd` ............................................................................. 9
- `extract_digitised` .................................................................. 10
- `filename` ................................................................................ 10
- `getExtracted` .......................................................................... 11
- `getVals` .................................................................................. 12
- `get_notDone_file_details` ......................................................... 12
- `grandMean` .............................................................................. 13
- `grandSD` .................................................................................. 14
- `group_scatter_extract` .............................................................. 15
- `histogram_extract` ................................................................ 15
- `import_menu` .......................................................................... 16
- `import_metaDigitise` ............................................................... 16
- `internal_digitise` .................................................................. 17
- `internal_redraw` .................................................................... 17
- `is.even` ................................................................................... 18
- `is.wholenumber` ..................................................................... 19
- `isNumeric` .............................................................................. 19
- `knownN` .................................................................................. 20
- `load_metaDigitise` .................................................................. 20
- `locator_mD` ............................................................................ 21
- `logAxes` .................................................................................. 21
- `MB_extract` ............................................................................ 22
- `metaDigitise` .......................................................................... 22
- `order_lists` .............................................................................. 24
- `plot.metaDigitise` .................................................................. 25
- `point_extraction` .................................................................... 25
- `print.metaDigitise` ................................................................. 26
- `print_cal_instructions` ............................................................. 26
- `process_data` .......................................................................... 27
- `process_new_files` ................................................................. 27
- `range_to_sd` ............................................................................ 28
- `redraw_calibration` ................................................................. 29
- `redraw_points` ....................................................................... 29
- `redraw_rotation` .................................................................... 30
- `rqm_to_mean` .......................................................................... 30
- `rqm_to_sd` ............................................................................. 31
- `setup_calibration_dir` .............................................................. 32
- `se_to_sd` ................................................................................ 32
- `single_MB_extract` ................................................................ 33
- `specify_type` .......................................................................... 33
- `summary.metaDigitise` ............................................................ 34
ask_variable

Description

asks user what variable(s) is depending on plot type

Usage

ask_variable(plot_type)

Arguments

plot_type

bulk_edit

Description

Function for bulk editing previous data extraction through ‘metaDigitise’

Usage

bulk_edit(dir, summary = TRUE, cex)

Arguments

dir
summary
cex

parent directory
logical; whether summary is returned
relative size of text and points in replotting

Author(s)

Joel Pick
### calibrate

**Description**

Converts x and y coordinates from original plot coords to actual coords using previous identified coordinates. Modified from digitise package.

**Usage**

```r
calibrate(raw_data, calpoints, point_vals, log_axes, ...)
```

**Arguments**

- `raw_data`: The raw data
- `calpoints`: The calibration points
- `point_vals`: The point values
- `log_axes`: whether x or y is logged
- `...`: further arguments passed to or from other methods

### cal_coords

**Description**

Prompts user to enter axis coordinates, and their values. Modified from the digitize package.

**Usage**

```r
cal_coords(plot_type, cex, ...)
```

**Arguments**

- `plot_type`: plot type
- `cex`: size of points
- `...`: further arguments passed to or from other methods.
Description

prints a vector as a number list of items with a certain number of columns

Usage

cat_matrix(x, cols)

Arguments

x vector
cols number of columns

Description

Transforms symmetrical confidence interval to standard deviation

Usage

CI95_to_sd(CI, n)

Arguments

CI Interval difference from the mean
n Sample Size

Value

Returns vector of standard deviations

Author(s)

Joel Pick

Examples

CI95_to_sd(CI = 2, n = 10)
convert_group_data

Description
Converts, pre-calibrated points clicked into a meaningful dataframe

Usage
convert_group_data(cal_data, plot_type)

Arguments
- cal_data: Calibrated data
- plot_type: The type of plot

convert_histogram_data

Description
Conversion of extracted data from histogram

Usage
convert_histogram_data(cal_data)

Arguments
- cal_data: The calibration data

delete_group

Description
Delete groups from scatterplots

Usage
delete_group(raw_data)

Arguments
- raw_data: data
Description

Function will gather important directory details about calibration files and figures needed for processing.

Usage

dir_details(dir)

Arguments

dir the path name to the directory / folder where the files are located

Author(s)

Daniel Noble - daniel.wa.noble@gmail.com

Examples

# temporary directory
tmp_dir <- tempdir()

setup_calibration_dir(paste0(tmp_dir, "/"))

# Simulate data
set.seed(103)
x <- rnorm(20,0,1)
y <- rnorm(20,0,1)
means <- c(mean(x), mean(y))
se <- c(sd(x)/sqrt(length(x))*1.96, sd(y)/sqrt(length(y))*1.96)

#Generate mock figures
png(filename = paste0(tmp_dir,"/mean_error.png"), width = 480, height = 480)
plot(means, ylim = c(min(means-sees)-0.1,max(means+sees)+0.1), xlim=c(0.5,2.5), xaxt="n", pch=19, cex=2, ylab="Variable +/- SE", xlab="Treatment", main="Mean Error")
arrows(1:length(means),means+sees,1:length(means), means-sees, code=3, angle=90, length=0.1)
axis(1,1:length(means), names(means))
dev.off()

png(filename = paste0(tmp_dir,"/boxplot.png"), width = 480, height = 480)
boxplot(x,y, main="Boxplot")
dev.off()

png(filename = paste0(tmp_dir,"/histogram.png"),width = 480, height = 480)
hist(c(x,y), xlab="variable", main="Histogram")
dev.off()

png(filename = paste0(tmp_dir,"/scatterplot.png"), width = 480, height = 480)
plot(x,y, main="Scatterplot")
dev.off()

# Obtain details on directory structure that are used for metaDigitise
data <- dir_details(tmp_dir)

---

**edit_group**

**Description**
Edit group points in scatterplots

**Usage**

```
edit_group(raw_data, group_id, calpoints, cex, ...)
```

**Arguments**

- `raw_data`: data
- `group_id`: group_id
- `calpoints`: The calibration points
- `cex`: point size
- `...`: other functions to pass to internal_redraw

---

**edit_metaDigitise**

**Description**
Function for editing previous data extraction through `metaDigitise`

**Usage**

```
edit_metaDigitise(object)
```

**Arguments**

- `object`: an R object of class `metaDigitise`

**Value**

Data.frame

**Author(s)**

Joel Pick
**enter_N**

**Description**

Enter sample sizes for a group

**Usage**

```
enter_N(raw_data, ...)
```

**Arguments**

- `raw_data`: raw_data
- `...`: Pass additional arguments

**Author(s)**

Joel Pick

---

**error_to_sd**

**Description**

Transforms error to standard deviation

**Usage**

```
error_to_sd(error, n, error_type = c("se", "CI95", "sd", NA))
```

**Arguments**

- `error`: some form of error
- `n`: Sample Size
- `error_type`: type of error measured

**Value**

Returns vector of standard errors

**Author(s)**

Joel Pick
**extract_digitised**

**Description**

Function for extracting the data from a metaDigitise list and creating either summary data or a list of the raw data.

**Usage**

```r
extract_digitised(list, summary = TRUE)
```

**Arguments**

- `list` A list of objects returned from metaDigitise
- `summary` A logical 'TRUE' or 'FALSE' indicating whether metaDigitise should print summary statistics from each figure and group.

**Value**

The function will return a data frame with the data across all the digitised files.

---

**filename**

**Description**

extracts filename from filepath

**Usage**

```r
filename(x)
```

**Arguments**

- `x` filepath
getExtracted

**Description**

Extracts data from a directory that has been previously digitised using metaDigitise()

**Usage**

getExtracted(dir, summary = TRUE)

**Arguments**

- **dir**
  The directory where figures have already been digitised. There

- **summary**
  Logical indicating whether summarised (default) or calibrated data should be returned.

**Value**

Returns a data frame (summary = TRUE) or a list with slots for each plot type (summary = FALSE)

**Examples**

```r
# Make some mock metaDigitise object
mock_metadig <- list(
  image_file = "/image.png",
  flip=FALSE, 
  rotate=0, 
  plot_type="mean_error", 
  variable="y", 
  calpoints = data.frame(x=c(0,0),y=c(0,100)), 
  point_vals = c(1,2), 
  entered_N=TRUE, 
  raw_data = data.frame(id=rep("control",2), 
x=c(60,60), 
y=c(75,50), 
n=rep(20,2)), 
knownN = NULL, 
error_type="sd", 
processed_data=data.frame( 
id=as.factor("control"), 
mean=1.5, 
error=0.25, 
n=20, 
variable="y", 
stringsAsFactors = FALSE) 
)
class(mock_metadig) <- 'metaDigitise'
```
# write image file to tmpdir()
dir <- tempdir()

# Setup directory as it would be if digitised images existed
setup_calibration_dir(dir)

# Save the digitised data
saveRDS(mock_metaDig, file = paste0(dir, "/caldat/", "image"))

#metaDigitise figures
data <- getExtracted(dir)

---

getVals

getVals

Description

Gets values needed to calibrate axis coordinated. Modified from the digitize package

Usage

getVals(calpoints, ...)

Arguments

calpoints Calibration points
...

further arguments passed to or from other methods.

---

get_notDone_file_details

get_notDone_file_details

Description

Function will get file information from the directory and the calibration files. It will also exclude files that have already been processed, as is judged by the match between file names in the calibration folder and the imported details object

Usage

get_notDone_file_details(dir)

Arguments

dir Path name to the directory / folder where the figure files are located.
**Value**

Returns a list containing details on the images names and their paths, the calibration file names (or files already completed) as well as the paths to these files.

**Author(s)**

Daniel Noble - daniel.wa.noble@gmail.com

**Examples**

```r
# temporary directory
tmp_dir <- tempdir()

# Simulate data
set.seed(103)
x <- rnorm(20, 0, 1)
y <- rnorm(20, 0, 1)
means <- c(mean(x), mean(y))
se <- c(sd(x)/sqrt(length(x))*1.96, sd(y)/sqrt(length(y))*1.96)

# Generate mock figures
png(filename = paste0(tmp_dir, "/mean_error.png"), width = 480, height = 480)
plot(means, ylim = c(min(means-ses), max(means+ses)), xlim=c(0.5, 2.5),
     xaxt="n", pch=19, cex=2, xlab="Variable +/- SE", ylab="Treatment", main="Mean Error")
arrows(1:length(means), means+ses, 1:length(means), means-ses, code=3, angle=90, length=0.1)
axis(1, at=1:length(means), labels=names(means))
dev.off()

png(filename = paste0(tmp_dir, "/boxplot.png"), width = 480, height = 480)
boxplot(x, y, main="Boxplot")
dev.off()

hist(c(x,y), main="variable", main="Histogram")
dev.off()

plot(x, y, main="Scatterplot")
dev.off()

# Obtain file names that are incomplete within the tmp directory
data <- get_notDone_file_details(tmp_dir)
```

**Description**

Pooled mean of a set of group means
Usage
grandMean(mean, n)

Arguments
- mean: Mean
- n: Sample size

Value
Returns vector of pooled mean

Author(s)
Joel Pick

Examples
grandMean(mean = 10, n = 30)

---

Description
Pooled standard deviation of a set of groups

Usage
grandSD(mean, sd, n, equal = FALSE)

Arguments
- mean: Mean
- sd: standard deviation
- n: Sample size
- equal: Logical: Whether to calculate pooled SD assuming groups have the same means (TRUE) or different means (FALSE)

Value
Returns vector of pooled mean

Author(s)
Joel Pick
**Examples**

```r
grandSD(mean = 10, sd = 3, n = 40)
```

---

**Description**

Extraction of data from scatterplots

**Usage**

```r
group_scatter_extract(edit = FALSE, raw_data = data.frame(), cex, ...)
```

**Arguments**

- `edit`: logical; whether in edit mode
- `raw_data`: raw data
- `cex`: point size
- `...`: arguments passed to internal_redraw

---

**Description**

Extraction of data from histograms

**Usage**

```r
histogram_extract(edit = FALSE, raw_data = data.frame(), calpoints, cex, ...)
```

**Arguments**

- `edit`: logical; whether in edit mode
- `raw_data`: raw data
- `calpoints`: The calibration points
- `cex`: point size
- `...`: arguments to pass to internal_redraw
Description
Imports metaDigitise() calibration files from a directory that is partially or fully digitised already

Usage
import_menu(dir, summary)

Arguments
- **dir**: The directory where figures have already been digitised
- **summary**: Logical indicating whether the imported data should be returned in summarised or processed form.

Value
Returns a list (summary = FALSE) or data frame (summary = TRUE)

Author(s)
Daniel Noble - daniel.wa.noble@gmail.com
internal_digitise

**Description**

Extracts points from a single figure and processes data

**Usage**

`internal_digitise(image_file, plot_type = NULL, cex)`

**Arguments**

- `image_file` Image file
- `plot_type` Type of plot from "mean_error","boxplot","scatterplot" or"histogram". Function will prompt if not entered by user.
- `cex` point size for replotting

**Value**

List of user inputs and transformed data from digitisation

**Author(s)**

Joel Pick

internal_redraw

**Description**

Redraws figure and extraction data

**Usage**

`internal_redraw(image_file, flip = FALSE, rotate = 0, plot_type = NULL, variable = NULL, cex = NULL, calpoints = NULL, point_vals = NULL, raw_data = NULL, rotation = TRUE, calibration = TRUE, points = TRUE, ...)`
Arguments

- `image_file`: Image filename
- `flip`: whether to flip figure
- `rotate`: how much to rotate figure
- `plot_type`: plot_type
- `variable`: variable
- `cex`: relative size of points and text
- `calpoints`: The calibration points
- `point_vals`: The point values
- `raw_data`: The raw data
- `rotation`: logical, should figure be rotated
- `calibration`: logical, should calibration be redrawn
- `points`: logical, should points be redrawn
- `...`: further arguments passed to or from other methods.

Description

Checks whether a integer is even

Usage

`is.even(x)`

Arguments

- `x`: integer value

Value

Logical (TRUE or FALSE) indicating whether value is an even number or not
Description

Checks whether value is a whole number

Usage

```r
is.wholenumber(x, tol = .Machine$double.eps^0.5)
```

Arguments

- `x` object to be tested
- `tol` tolerance

Value

Logical value (TRUE or FALSE)

Description

Checks whether a character is a number

Usage

```r
isNumeric(x)
```

Arguments

- `x` character to be tested

Value

Logical (TRUE or FALSE) indicating whether value is numeric or not
knownN  knownN

Description

prints a vector as a number list of items with a certain number of columns

Usage

knownN(plot_type, processed_data, knownN = NULL, ...)

Arguments

plot_type  plot type
processed_data  raw_data
knownN  previously entered N
...  arguments from other calls

load_metaDigitise

Description

Loads metaDigitise calibration / data files from a directory containing a set of figures that are partially or fully digitised already.

Usage

load_metaDigitise(doneCalFiles, names)

Arguments

doneCalFiles  The metaDigitise objects that have already been completed in the directory
names  The names of the finished metaDigitise objects

Value

Returns a list of metaDigitised objects that have already been completed

Author(s)

Daniel Noble - daniel.wa.noble@gmail.com
**Description**

Wrapper function for locator, with more control over point size etc

**Usage**

`locator_mD(nPoints = 1, line = TRUE, cex = 1, col = "red", ...)`

**Arguments**

- `nPoints`: number of points in a sequence
- `line`: logical; plot lines between points
- `cex`: size of points
- `col`: colour of points
- `...`: further arguments passed to or from other methods.

**Value**

Plots clicked points, and returns their x,y coordinates as a data.frame

---

**Description**

Ask user for information about whether axes are on log scale

**Usage**

`logAxes()`

**Arguments**

`...`: further arguments passed to or from other methods.
### MB_extract

**Description**

Extraction of data from boxplots of mean_error plots, from multiple groups

**Usage**

```r
MB_extract(edit = FALSE, plot_type, entered_N, raw_data = data.frame(), cex, ...)
```

**Arguments**

- `edit`: logical; whether in edit mode
- `plot_type`: The type of plot
- `entered_N`: ask for sample sizes?
- `raw_data`: raw data
- `cex`: point size
- `...`: further arguments to MB_extract

---

### metaDigitise

**Description**

Single or batch processing of figures with .png, .jpg, .tiff, .pdf extensions within a set directory. `metaDigitise()` consolidates the data and exports the data for each image and image type. It can also summarise the data, provide the raw data (if scatterplots) and automatically imports previously finished data and merges it with newly digitised data. `metaDigitise()` also allows users to check their calibration along with editing previous digitisations.

**Usage**

```r
metaDigitise(dir, summary = TRUE, cex = 1)
```

**Arguments**

- `dir`: the path name to the directory / folder where the files are located
- `summary`: whether the digitised data should be returned as a summary (TRUE) or as a concatenated list of similar types.
- `cex`: relative size of points and text in replotting of digitisation. Default is 1.
Details

`metaDigitise()` can be used on a directory with a whole host of different figure types (mean and error, scatter plots, box plots and histograms) and file types (.jpeg, .png, .tiff, .pdf). There are three major options provided to users:

If the "1: Process new images" option is chosen, it will automatically cycle through all figures not already completed within a directory in order, prompting the user for specific information as they go. At the end of each figure users will be asked if they would like to continue or not, providing flexibility to leave a job should they need to. As figures are digitised it will automatically write `metaDigitise()` object files (in .RDS format containing processed and calibration data along with directory and file details), into a special caldat/ folder within the directory. Importantly, as new files are added to a directory that has already been "completed", `metaDigitise()` will recognize these unfinished files and only cycle through the digitisation of these new files. This easily allows users to pick up from where they left off. It will also automatically re-merge completed figure with any newly digitised figures at the end of this process keeping everything together throughout the process.

If the "2: Import existing data" is chosen, all existing files that have already been digitised will be automatically imported from the given directory.

Finally, `metaDigitise` is built for ease of editing and reproducibility in mind. Hence, if "3: Edit existing data" is chosen by the user then users will have the options to "1: Cycle through images" (that are complete), overlaying digitisations with each figure and asking whether they would like to edit each figure or "2: Choose specific file to edit" allowing editing for a specific file. Here a list of all files are provided and the user simply needs to pick the one in the console they would like to view. Alternatively, the "3: Enter previously omitted sample sizes" option allows the user to go back and enter sample sizes that they may not have had on hand at the time of digitisation. This means that, so long as the caldat/ folder along with respective images are maintained, anyone using `metaDigitise()` can simply import existing digitisations, modify them and fix them. This folder can then be shared with colleagues to allow them to reproduce any data extraction.

Value

A data frame or list containing the raw digitised data or the processed, summary statistics from the digitised data

Author(s)

Joel Pick - joel.l.pick@gmail.com

Daniel Noble - daniel.wa.noble@gmail.com

Examples

```r
# temporary directory
tmp_dir <- tempdir()

# Simulate data
set.seed(103)
x <- rnorm(20, 0, 1)
y <- rnorm(20, 0, 1)
```
order_lists

Description

Will re-order the processed data such that similar types of data are organised into a single list defined by their plot type.

Usage

order_lists(list, plot_types)

Arguments

list The list of metaDigitise objects that have already been finished within the caldat/ folder
plot_types The list of plot types extracted from metaDigitised objects

Value

Returns a list ordered by the plot type

```r
means <- c(mean(x), mean(y))
ses <- c(sd(x)/sqrt(length(x))*1.96, sd(y)/sqrt(length(y))*1.96)

# Generate mock figures
png(filename = paste0(tmp_dir, "/mean_error.png"), width = 480, height = 480)
plot(means, ylim = c(min(means-ses)-0.1, max(means+ses)+0.1), xlim=c(0.5, 2.5),
     xaxt="n", pch=19, cex=2, ylab="Variable +/- SE", xlab="Treatment", main="Mean Error")
arrows(1:length(means), means+ses, 1:length(means), means-ses, code=3, angle=90, length=0.1)
axis(1,1:length(means), names(means))
dev.off()

png(filename = paste0(tmp_dir, "/boxplot.png"), width = 480, height = 480)
boxplot(x, y, main="Boxplot")
dev.off()

hist(c(x, y), main="variable", main="Histogram")
dev.off()

# metaDigitise figures
## Not run:
data <- metaDigitise(tmp_dir)

## End(Not run)
```
plot.metaDigitise

Author(s)
Daniel Noble - daniel.wa.noble@gmail.com

Description
Re-plots figure and extraction data

Usage

## S3 method for class 'metaDigitise'
plot(x, cex = NULL, ...)

Arguments

x an R object of class `metaDigitise`
cex size of points
... further arguments passed to or from other methods.

Author(s)
Joel Pick

point_extraction

Description
Extracts or edits point of a digitisation

Usage

point_extraction(object, edit = FALSE)

Arguments

object Object
edit Logical (TRUE or FALSE) indicating whether a point would like to be edited
print.metaDigitise  

Description
Print method for class 'metaDigitise'

Usage
## S3 method for class 'metaDigitise'
print(x, ...)

Arguments
x an R object of class 'metaDigitise'
... further arguments passed to or from other methods.

Author(s)
Joel Pick

print_cal_instructions

Description
Prints instructions for calibration. Modified from the digitize package

Usage
print_cal_instructions(plot_type, ...)

Arguments
plot_type plot type
... further arguments passed to or from other methods.
process_data

Description
 Processes points clicked into a meaningful dataframe

Usage
 process_data(object)

Arguments
 object object from metaDigitise

process_new_files

Description
 Batch processes image files within a set directory, consolidates the data and exports the data for each image and type

Usage
 process_new_files(dir, summary = TRUE, cex)

Arguments
 dir the path name to the directory / folder where the files are located
 summary summary = TRUE or FALSE is most relevant as it will print a simple summary statistics that are the same across all files
 cex relative size of points and text in replotting of digitisation.

Author(s)
 Joel Pick - joel.l.pick@gmail.com
 Daniel Noble - daniel.wa.noble@gmail.com
Examples

```r
# temporary directory
tmp_dir <- tempdir()

# Simulate data
set.seed(103)
x <- rnorm(20, 0, 1)
y <- rnorm(20, 0, 1)
means <- c(mean(x), mean(y))
sses <- c(sd(x)/sqrt(length(x))*1.96, sd(y)/sqrt(length(y))*1.96)

# Generate mock mean error plot
png(filename = paste0(tmp_dir, "/mean_error.png"), width = 480, height = 480)
plot(means, ylim = c(min(means-sses)-0.1, max(means+sses)+0.1), xlab="treatment", main="Mean Error")
axis(1, length=length(means), names=names(means))
dev.off()
```

### Not run:

# metaDigitise figures
data <- process_new_files(paste0(tmp_dir, "/"), summary = TRUE, cex = 2)

### End(Not run)

---

**range_to_sd**

**Description**

Converts a range to a standard deviation

**Usage**

```r
range_to_sd(min, max, n)
```

**Arguments**

- `min` Minimum value
- `max` Maximum value
- `n` Sample size

**Value**

Returns vector of standard deviation
**redraw_calibration**

**Author(s)**

Joel Pick

**Examples**

```
range_to_sd(min = 3, max = 8, n = 40)
```

---

**Description**

plots calibration data on graph

**Usage**

```
redraw_calibration(plot_type, variable, calpoints, point_vals, image_details, cex)
```

**Arguments**

- `plot_type`: plot_type
- `variable`: variable
- `calpoints`: The calibration points
- `point_vals`: The point values
- `image_details`: image_details
- `cex`: relative size of points and text

---

**redraw_points**

**Description**

plots clicked data on graph

**Usage**

```
redraw_points(plot_type, raw_data, image_details, cex)
```

**Arguments**

- `plot_type`: plot_type
- `raw_data`: The raw data
- `image_details`: image_details
- `cex`: relative size of points and text
redraw_rotation

Description
Rotates/flips imported figures

Usage
redraw_rotation(image, flip, rotate)

Arguments
image               Image object from magick::image_read
flip                whether to flip figure
rotate              how much to rotate figure

rqm_to_mean

Description
Calculate the mean from the box plots

Usage
rqm_to_mean(min, LQ, median, UQ, max, n)

Arguments
min     Minimum value
LQ      Lower 75th quartile
median  Median
UQ      Upper 75th quartile
max     Maximum value
n       Sample size

Value
Returns vector of mean

Author(s)
Joel Pick
**Examples**

```
rqm_to_mean(min = 2, LQ = 3, median = 5, UQ = 6, max = 9, n = 30)
```

---

**Description**

Calculate the standard deviation from box plots

**Usage**

```
rqm_to_sd(min, LQ, UQ, max, n)
```

**Arguments**

- `min` Minimum value
- `LQ` Lower 75th quartile
- `UQ` Upper 75th quartile
- `max` Maximum value
- `n` Sample size

**Value**

Returns vector of standard deviation

**Author(s)**

Joel Pick

**Examples**

```
rqm_to_sd(min = 2, LQ = 3, UQ = 6, max = 9, n = 30)
```
setup_calibration_dir

Description
Function will check whether the calibration directory has been setup and if not, create one.

Usage
setup_calibration_dir(dir)

Arguments
- dir: Path name to the directory / folder where the files are located.

Value
Returns a caldat/ folder within the directory where all metaDigitise objects are stored.

Author(s)
Daniel Noble - daniel.wa.noble@gmail.com

Examples

```r
# temporary directory
tmp_dir <- tempdir()

#Create the calibration folder in the directory specified that is used to store files.
setup_calibration_dir(paste0(tmp_dir, "/"))
```

se_to_sd

Description
Transforms standard error to standard deviation

Usage
se_to_sd(se, n)
Arguments

- **se**: Standard Error of the mean
- **n**: Sample Size

Value

Returns vector of standard errors

Author(s)

Joel Pick

Examples

```r
se_to_sd(se = 5, n = 10)
```

---

**Description**

Takes points user defined points from a single group mean error plot or boxplot, in a set order, and returns them.

**Usage**

```r
single_MB_extract(plot_type, cex)
```

**Arguments**

- **plot_type**: Type of plot
- **cex**: point size

---

**Description**

Function that allows user to interface with function to specific each type of plot prior to digitising

**Usage**

```r
specify_type()
```
Value

The function will return the type of plot specified by the user and feed this argument back into metaDigitise

Author(s)

Daniel Noble - daniel.wa.noble@gmail.com
Joel Pick - joel.l.pick@gmail.com

Summary method for class `metaDigitise`

## Usage

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

### Arguments

- `object` an R object of class `metaDigitise`
- `...` further arguments passed to or from other methods.

### Value

Data.frame

Author(s)

Joel Pick

## Description

asks user for base of logarithm, accept numeric or "e"

## Usage

```r
user_base(...)```

### Arguments

- `...` arguments passed to other functions
user_calibrate

Description
Gets values needed to calibrate axis coordinated. Modified from the digitize package

Usage
user_calibrate(object)

Arguments
object metaDigitise object

user_count

Description
asks user for count

Usage
user_count(question)

Arguments
question question

user_numeric

Description
asks user for numeric

Usage
user_numeric(question)

Arguments
question question
**Description**

asks user for option from specified list

**Usage**

`user_options(question, allowed_answers)`

**Arguments**

- `question`  
- `allowed_answers`

**Description**

Rotates/flips imported figures according to user input, in order to align them properly. Asks the user after each change if further alteration is required

**Usage**

`user_rotate_graph(image_file)`

**Arguments**

- `image_file`  
  Image filename
Description

asks user for option from specified list

Usage

user_unique(question, previous_answers)

Arguments

question
previous_answers

allowed answers
Index

ask_variable, 3
bulk_edit, 3
cal_coords, 4
calibrate, 4
cat_matrix, 5
CI95_to_sd, 5
convert_group_data, 6
convert_histogram_data, 6
delete_group, 6
dir_details, 7
edit_group, 8
edit_metadigitise, 8
enter_N, 9
error_to_sd, 9
extract_digitised, 10
filename, 10
get_notDone_file_details, 12
getExtracted, 11
getVals, 12
grandMean, 13
grandSD, 14
group_scatter_extract, 15
histogram_extract, 15
import_menu, 16
import_metadigitise, 16
internal_digitise, 17
internal_redraw, 17
is:even, 18
is:wholenumber, 19
isNumeric, 19
knownN, 20
load_metadigitise, 20
locator_md, 21
logAxes, 21
MB_extract, 22
metaDigitise, 22
order_lists, 24
plot.metaDigitise, 25
point_extraction, 25
print.metaDigitise, 26
print_cal_instructions, 26
process_data, 27
process_new_files, 27
range_to_sd, 28
redraw_calibration, 29
redraw_points, 29
redraw_rotation, 30
rqm_to_mean, 30
rqm_to_sd, 31
se_to_sd, 32
setup_calibration_dir, 32
single_MB_extract, 33
specify_type, 33
summary.metaDigitise, 34
user_base, 34
user_calibrate, 35
user_count, 35
user_numeric, 35
user_options, 36
user_rotate_graph, 36
user_unique, 37