# Package ‘modeldiagramR’

May 9, 2026

**Type** Package

**Title** Generate Model Diagrams for Linear Mixed Effect Models

**Version** 0.2.1

**Description** Generates 'DiagrammeR' model diagrams for hierarchical linear mixed effects models. Details can be found in Linse (2026) <doi:10.6339/26-JDS1222>.

**Depends** R (>= 4.1),

**Imports** stats, utils, methods, stringr, dplyr, tidyr, tidyselect, magrittr, tibble, gtools, forcats, nlme, DiagrammeR

**Suggests** ggplot2, patchwork, ggthemes, lme4, lmerTest, knitr, viridis, DiagrammeRsvg, webshot, FielDHub, rsvg

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.3

**URL** `https://github.com/glinse-stat/modeldiagramr`

**BugReports** `https://github.com/glinse-stat/modeldiagramr/issues`

**NeedsCompilation** no

**Author** Greta Linse [aut, cre],
Mark Greenwood [aut]

**Maintainer** Greta Linse <`greta.linse@montana.edu`>

**Repository** CRAN

**Date/Publication** 2026-04-15 13:00:23 UTC

## Contents

---

`cetaceans`

## *Data from a repeated measures study on age by tooth staining methods of stranded Cetaceans*

---

### Description

Results of a study comparing three staining methods to determine age of dolphins from six species of cetaceans from two different locations (Spain and Scotland). It is unclear how the numeric values of sex are coded so they are left as integers. This dataset has been cleaned to remove the dolphin with undetermined sex (DolphinID = 10).

### Usage

```
cetaceans
```

### Format

a `data.frame` with 177 observations on 11 variables from 59 dolphins:

**DolphinID** Dolphin ID from 1 to 61 (excluding 10 and 16)

**Species** Species of cetacean (6 species)

**Age** Age determined by staining method from a tooth, numeric

**Sex** Sex of dolphin, integer

**Stain** Treatment, staining method (Mayer, Elrich, Toluidine), categorical

**Location** Location of stranding (Scotland, Spain), categorical

**fSpecies** Factor version of Species variable

**fDolphinID** Factor version of DolphinID

**fStain** Factor version of Stain

**fLocation** Factor version of Location

**fSex** Factor version of Sex

### Details

Additional variables include sex of the animal, location of the stranding, and stain (Mayer Haematoxylin, Ehlrich Haematoxylin, and Toluidine Blue).

### Source

https://www.highstat.com/Books/Book2/ZuurDataMixedModelling.zip

### References

Zuur AF, Ieno EN, Walker N, Saveliev AA, and Smith GM. 2009. Mixed effects models and extensions in ecology with R. Statistics for Biology and Health. Springer New York, p. 460. DOI: 10.1007/978-0-387-87458-6

### Examples

```
library(dplyr)
data("cetaceans")


# Exploratory Data Plots
library(ggplot2)
library(viridis)
p1 <- cetaceans %>%
  ggplot(aes(x = DolphinID, y = Age, group=fDolphinID)) +
  geom_boxplot(aes(color=fSpecies)) +
  scale_color_viridis_d(end = 0.8) +
  theme_bw() +
  theme(legend.position="none")
p2 <- cetaceans %>%
  ggplot(aes(x = fSpecies, y = Age, group=fSpecies)) +
  geom_boxplot(aes(color = fSpecies)) +
  scale_color_viridis_d(end = 0.8) +
  theme_bw() +
  theme(axis.text.x = element_text(angle=90),
        legend.position="none")
p3 <- cetaceans %>% ggplot(aes(x = fSpecies, y = Age)) +
  geom_boxplot(aes(color = fSpecies)) +
  scale_color_viridis_d(end = 0.8) +
  facet_wrap(~fLocation)+
  theme_bw() +
  theme(axis.text.x = element_text(angle=90))

library(patchwork)
(p1) / (p2) / (p3)

# Simple model for testing
library(nlme)
lme_simple <- lme(Age ~ fSex*fStain*fLocation,
                  random = ~ 1|fSpecies/fDolphinID,
                  data = cetaceans)
model_diagram(lme_simple)


# Intended full model
library(lme4)
library(lmerTest)
lmer1 <- lmer(Age ~ fSex*fStain*fLocation + (1|fSpecies/fDolphinID), data = cetaceans)
summary(lmer1)
model_diagram(lmer1)
```

---

`combinedtattoo` *Data from a repeated measures study on tattoo sweat rates*

## Description

Results of a study of 10 subjects on sweat rates and sodium concentrations, paired between two locations on each subject I digitized the plotted data to create the provided data set, results do not exactly match the original results.

## Usage

```
combinedtattoo
```

## Format

a `data.frame` with 20 observations on 10 variables from 10 subjects:

**Subject** Subject ID from 1 to 10 0

**SweatRate** Rate of sweat from a location, mg per cm-squared per min

**Tat_not** Tattoo location or Not

**Na_conc** Na concentration, mMol/L

**Height_cm** Height of subject in cm

**Weight_kg** Weight of subject in kg

**Age_yr** Age of subject in years

**Side** Side of subject, left or right

**Location** Location of tattoo, categorical

**Tattoo_Age** Age of tattoo (replicated both subject observations), years

## Details

Subject demographics are assumed to be correctly matched but were merged based on the information in a plot and in a separate table

## Source

Digitized version of published Figures 1 and 2 combined with Table 1 (does not exactly match published numerical summaries!)

## References

Luetkemeier, M., Hanisko, J., and K. Aho (2017) Skin Tattoos Alter Sweat Rate and Na+ Concentration. Medicine & Science in Sports & Exercise 49(7):p 1432-1436. DOI: 10.1249/MSS.0000000000001244

## Examples

```
library(dplyr)
data("combinedtattoo")
combinedtattoo <- combinedtattoo %>%
  mutate(SubjectF = factor(Subject))
```

```
# Exploratory Data Plots
library(ggthemes)
library(ggplot2)
library(viridis)
p1 <- combinedtattoo %>%
  ggplot(aes(x = Tat_not, y = SweatRate, group = SubjectF)) +
    geom_line(aes(color = SubjectF)) +
    scale_color_viridis_d(end = 0.8)
p2 <- combinedtattoo %>%
  ggplot(aes(x = Tat_not, y = Na_conc, group = SubjectF)) +
    geom_line(aes(color = SubjectF)) +
    scale_color_viridis_d(end = 0.8)
p3 <- combinedtattoo %>%
  ggplot(aes(x = Weight_kg, y = SweatRate)) +
    geom_point(aes(color = Subject)) +
    scale_color_viridis_c(end = 0.8) +
    geom_smooth(method = "lm") +
    facet_wrap(~Tat_not)

library(patchwork)
(p1 + p2) / (p3)


library(lmerTest)

lmer1 <- lmer(SweatRate ~ Tat_not + (1|Subject),
              data = combinedtattoo)
summary(lmer1)

model_diagram(lmer1)


lmerFL <- lmer(SweatRate ~ Tat_not*Weight_kg + (1|Subject),
               data = combinedtattoo)
summary(lmerFL)

model_diagram(lmerFL)
```

---

`md` *Diagram elements*

## Description

Similar to `ggplot2::ggplot()`'s `ggplot2::theme()` system, the `md_` functions specify the display of how the node components of the diagram are drawn.

- `md_color()`: border of the node.
- `md_fill()`: fill color of the node.
- `md_fontColor()`: font color of the text in the nodes.

## Usage

```
md_color(diagram = "gray25", random = "gray25", fixed = "gray25")

md_fill(diagram = "aliceblue", random = "aliceblue", fixed = "darkseagreen1")

md_fontColor(diagram = "black", random = "black", fixed = "black")
```

## Arguments

| | |
|---|---|
| `diagram` | Specifies the outline color, fill color, or font color for the elements in the measurement diagram circles (hierarchical diagram). Default is `"gray25"` for (`md_color()`), `"aliceblue"` for (`md_fill()`), and `"black"` for (`md_fontColor()`). |
| `random` | Specifies the outline color, fill color, or font color for the elements in the random effect variable boxes. Default is `"gray25"` for (`md_color()`), `"aliceblue"` for (`md_fill()`), and `"black"` for (`md_fontColor()`). |
| `fixed` | Specifies the outline color, fill color, or font color for the elements in the fixed effect variable boxes. Default is `"gray25"` for (`md_color()`), `"darkseagreen1"` for (`md_fill()`), and `"black"` for (`md_fontColor()`). |

## Value

A list object containing color specifications for the border of the nodes.

A list object containing color specifications for the fill of the nodes.

A list object containing color specifications for the font color of the nodes.

## Examples

```
# merMod object example
library(lme4)

sleepstudy_lmer <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)
summary(sleepstudy_lmer)
model_diagram(sleepstudy_lmer,
 nodeColors = md_color(diagram="steelblue", random="steelblue", fixed="darkolivegreen1"),
 nodeFillColors = md_fill(diagram="aliceblue", random="aliceblue", fixed="darkseagreen1"),
  nodeFontColors = md_fontColor(diagram="black", random="blue", fixed="black"))
```

---

# `model_diagram` *Hierarchical model diagramming*

## Description

`model_diagram()` takes a hierarchical nested model and returns a DiagrammeR object visualizing the fixed and random effects structure.

### Usage

```
model_diagram(
  modelObject,
  filePath = NULL,
  fileType = "PNG",
  width = 800,
  height = 1600,
  includeSizes = TRUE,
  includeLabels = TRUE,
  orientation = "vertical",
  scaleFontSize = 1,
  shiftFixed = 0,
  shiftRandom = 0,
  nodeColors = md_color(diagram = "gray25", random = "gray25", fixed = "gray25"),
  nodeFillColors = md_fill(diagram = "aliceblue", random = "aliceblue", fixed =
    "darkseagreen1"),
  nodeFontColors = md_fontColor(diagram = "black", random = "black", fixed = "black"),
  exportObject = 1
)
```

### Arguments

| | |
|---|---|
| `modelObject` | Input model. Either a lme or merMod (including glmerMod) object with a nested random effects structure. |
| `filePath` | Optional. Path to a location to export the diagram. Default is `NULL`. |
| `fileType` | Optional. File type to export the diagram. Default is `"PNG"`. |
| `width` | Optional. Width of diagram in pixels. Default is `800`. |
| `height` | Optional. Height of diagram in pixels. Default is `1600`. |
| `includeSizes` | Optional. Include group sizes in random effect labels. Default is `TRUE`. |
| `includeLabels` | Optional. Include labels for the model diagram components. Default is `TRUE`. |
| `orientation` | Optional. Orientation of the diagram, either vertically or horizontally. Options are `"vertical"`, `"horizontal"`. Default is `"vertical"`. |
| `scaleFontSize` | Optional. Proportional font size adjustment for model diagram component, fixed effect, and random effect labels. Multiplies these label font sizes by the specified amount. Default is `1`. |
| `shiftFixed` | Optional. Additive x-axis adjustment for fixed effect labels, only used when `orientation == "horizontal"`. Default is `0`. |
| `shiftRandom` | Optional. Additive x-axis adjustment for random effect labels, only used when `orientation == "horizontal"`. Default is `0`. |
| `nodeColors` | Optional. Function specifying the colors (`md_color()`) for the outline of the nodes. Components can be specified individually (`diagram`, `random`, and `fixed`). |
| `nodeFillColors` | Optional. Function specifying the colors (`md_fill()`) for the fill color of the nodes. Components can be specified individually (`diagram`, `random`, and `fixed`). |

| | |
|---|---|
| `nodeFontColors` | Optional. Function specifying the colors (`md_fontColor()`) for the font color of text in the nodes. Components can be specified individually (`diagram`, `random`, and `fixed`). |
| `exportObject` | Optional. Object(s) to return, 1 for a rendered DiagrammeR graph as an SVG document, 2 for a `dgr_graph` object, and 3 returns a list containing both objects. Default is 1. |

## Details

NOTE: When including this function in an RMD document and knitting to PDF, the graphic size variables `width` and `height` do not currently work. It is recommended that instead the image is exported to a file such as a PDF and then reimported to the document. See the examples below.

## Value

A rendered DiagrammeR graph as an SVG document (default), or a `dgr_graph` object, or a list containing both (`dgr_graph_obj`, `rendered_graph_obj`).

## References

Greta M. Linse, Mark C. Greenwood, Ronald K. June, *Data-Driven Model Structure Diagrams for Hierarchical Linear Mixed Models*, J. data sci.(2026), 1-21, DOI 10.6339/26-JDS1222

## Examples

```
# merMod object example
library(lme4)

sleepstudy_lmer <- lmer(Reaction ~ Days + (Days | Subject), sleepstudy)
summary(sleepstudy_lmer)
model_diagram(sleepstudy_lmer)

# lme object example
library(nlme)

sleepstudy_lme <- lme(Reaction ~ Days, random=~Days|Subject, data=sleepstudy)
summary(sleepstudy_lme)
model_diagram(sleepstudy_lme)


library(rsvg) # required to produce a PDF file
temp_path <- tempfile("sleepstudy_lmer_modeldiagram", fileext=".pdf")
# Knitting to PDF example
model_diagram(sleepstudy_lmer,
               filePath= temp_path,
               fileType="PDF")
```

---

## SSPD3 *Simulated data for a split-split-plot experimental design.*

---

### Description

The data is simulated using `simulate_SSPD3_data()`

### Usage

```
SSPD3
```

### Format

a `data.frame` with 120 observations on 13 variables:

**ID** Observation/split-split-plot identification number, numeric

**LOCATION** One of three locations "A", "B", and "C", a blocking variable, categorical

**PLOT** Plot identification variable, one for each location and whole plot combination, numeric

**REP** Replication number, all 1 as there was no replication, integer

**WHOLE_PLOT** Treatment, whether or not the plot was irrigated "IRR_NO" for not irrigated and "IRR_YES" for irrigated, categorical

**SPLIT_PLOT** Treatment, fungicide applied to split-plot, 4 fungicides and one control, "Fung1", ..., "Fung4", and "NFung", categorical

**SPLIT_SPLIT_PLOT** Treatment, variety of bean seeded in split-split-plot, 4 varieties, "Beans1", ..., "Beans4", categorical

**TRT_COMB** Treatments combined over WHOLE_PLOT, SPLIT_PLOT, and SPLIT_SPLIT_PLOT

**RESP** Simulated response, not based on a meaningful input values, numeric

**LocationF** Factor version of LOCATION

**Irrigation** Factor version of WHOLE_PLOT treatment

**Fungicide** Factor version of SPLIT_PLOT treatment

**Variety** Factor version of SPLIT_SPLIT_PLOT treatment

### References

Murillo D, Gezan S (2024). *FielDHub: A Shiny App for Design of Experiments in Life Sciences*. R package version 1.4.2, https://CRAN.R-project.org/package=FielDHub.

### Examples

```
data("SSPD3")

# Simple model for testing
library(nlme)
lme_simple <- lme(RESP ~ Fungicide + Variety, random=~1|LOCATION/WHOLE_PLOT,
```

```
                     data=SSPD3)
model_diagram(lme_simple)


# Intended full model
aov_results <- aov(RESP ~ Irrigation*Fungicide*Variety + Error(LOCATION/WHOLE_PLOT/SPLIT_PLOT),
                   data=SSPD3)
summary(aov_results)
library(lme4)
library(modeldiagramR)
lmer_results <- lmer(RESP ~ Irrigation*Fungicide*Variety + (1|LOCATION/WHOLE_PLOT/SPLIT_PLOT),
                     data=SSPD3)
summary(lmer_results)

anova(lmer_results)

model_diagram(lmer_results)

model_diagram(lmer_results, width = 800, height=400, orientation="horizontal",
              shiftFixed = 2, shiftRandom = 5, scaleFontSize = 2)
```

# Index