

# Package ‘flowchart’

May 10, 2024

**Type** Package

**Title** Tidy Flowchart Generator

**Version** 0.3.0

**Maintainer** Pau Satorra <psatorra@igtp.cat>

**Description** Creates participant flow diagrams directly from a dataframe. Representing the flow of participants through each stage of a study, especially in clinical trials, is essential to assess the generalisability and validity of the results. This package provides a set of functions that can be combined with a pipe operator to create all kinds of flowcharts from a data frame in an easy way.

**License** MIT + file LICENSE

**BugReports** <https://github.com/bruigtp/flowchart/issues>

**Encoding** UTF-8

**LazyData** true

**Imports** Gmisc, grid, tidyr, dplyr, purrr, stringr, tibble, tidyselect, rlang, grDevices

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Depends** R (>= 4.1.0)

**RoxygenNote** 7.2.3

**URL** <https://bruigtp.github.io/flowchart/>

**NeedsCompilation** no

**Author** Pau Satorra [aut, cre],  
João Carmezim [aut],  
Natàlia Pallarès [aut],  
Cristian Tebé [aut]

**Repository** CRAN

**Date/Publication** 2024-05-10 12:50:02 UTC

## R topics documented:

as_fc . . . . .	2
clinic_patient . . . . .	3
clinic_visit . . . . .	4
fc_draw . . . . .	5
fc_export . . . . .	6
fc_filter . . . . .	7
fc_merge . . . . .	9
fc_modify . . . . .	10
fc_split . . . . .	11
fc_stack . . . . .	12
safo . . . . .	13

<b>Index</b>	<b>15</b>
--------------	-----------

---

as_fc	<i>as_fc</i>
-------	--------------

---

### Description

This function allows to initialize a flow chart given any database. It will create a fc object showing the number of rows of the database. If a database is not available, the user can instead directly enter the number of rows in the study.

### Usage

```
as_fc(
  .data = NULL,
  N = NULL,
  label = "Initial dataframe",
  text_pattern = "{label}\n{N}",
  just = "center",
  text_color = "black",
  text_fs = 8,
  bg_fill = "white",
  border_color = "black",
  hide = FALSE
)
```

### Arguments

.data	Data frame to be initialised as a flowchart.
N	Number of rows of the study in case '.data' is NULL.
label	Character with the text that will be shown in the box.
text_pattern	Structure that will have the text in the box. It recognizes label, n, N and perc within brackets. For default it is "{label}\n{N}".

just	Justification for the text: left, center or right. Default is center.
text_color	Color of the text. It is black by default.
text_fs	Font size of the text. It is 8 by default.
bg_fill	Box background color. It is white by default.
border_color	Box border color. It is black by default.
hide	Logical value to hide the initial box or not. Default is FALSE. hide = TRUE can only be combined with fc_split().

**Value**

List with the dataset and the initialized flowchart parameters.

**Examples**

```
clinic_patient |>
as_fc(label = "All patients") |>
fc_draw()
```

---

clinic_patient	<i>Simulated clinical trial information by patient Simulated dataset of a clinical trial comparing some biomarker values between two randomized groups (control/treatment). This dataset contains the information by patient. It is equivalent to the dataset clinic_visit, which has the same information by visit.</i>
----------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

**Description**

Simulated clinical trial information by patient Simulated dataset of a clinical trial comparing some biomarker values between two randomized groups (control/treatment). This dataset contains the information by patient. It is equivalent to the dataset clinic\_visit, which has the same information by visit.

**Usage**

```
data(clinic_patient)
```

**Format**

A data frame with 230 rows and 8 columns

**id:** Identifier of each patient.

**age:** Age values.

**consent:** The patient has signed the informed consent?

**group:** Randomized group: control/treatment.

**n\_visits:** Number of follow-up visits.

**marker\_alt:** Is the marker altered in any of the visits?

## Examples

```
clinic_patient |>
  as_fc(label = "Available patients") |>
  fc_filter(age >= 18 & consent == "Yes", label = "Patients included", show_exc = TRUE) |>
  fc_split(group) |>
  fc_filter(n_visits == 2, label = "Two visits available", show_exc = TRUE) |>
  fc_split(marker_alt, label = c("Marker not altered", "Marker altered")) |>
  fc_draw()
```

---

clinic_visit	<i>Simulated clinical trial information by visit Simulated dataset of a clinical trial comparing some biomarker values between two randomized groups (control/treatment). This dataset contains the information by visit. It is equivalent to the dataset clinic_patient, which has the same information by patient.</i>
--------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

## Description

Simulated clinical trial information by visit Simulated dataset of a clinical trial comparing some biomarker values between two randomized groups (control/treatment). This dataset contains the information by visit. It is equivalent to the dataset clinic\_patient, which has the same information by patient.

## Usage

```
data(clinic_visit)
```

## Format

A data frame with 411 rows and 8 columns

**id:** Identifier of each patient.

**age:** Age values.

**consent:** The patient has signed the informed consent?

**group:** Randomized group: control/treatment.

**visit:** Number of the corresponding follow-up visit.

**marker:** Marker value in the visit.

## Examples

```
clinic_visit |>
  dplyr::filter(!is.na(group)) |>
  as_fc(label = "Number of visits") |>
  fc_split(group) |>
  fc_draw()
```

---

`fc_draw`*fc\_draw*

---

## Description

This function allows to draw the flowchart from a fc object.

## Usage

```
fc_draw(  
  object,  
  arrow_angle = 30,  
  arrow_length = grid::unit(0.1, "inches"),  
  arrow_ends = "last",  
  arrow_type = "closed"  
)
```

## Arguments

<code>object</code>	fc object that we want to draw.
<code>arrow_angle</code>	The angle of the arrow head in degrees, as in 'arrow'.
<code>arrow_length</code>	A unit specifying the length of the arrow head (from tip to base), as in 'arrow'.
<code>arrow_ends</code>	One of "last", "first", or "both", indicating which ends of the line to draw arrow heads, as in 'arrow'.
<code>arrow_type</code>	One of "open" or "closed" indicating whether the arrow head should be a closed triangle, as in 'arrow'.

## Value

Invisibly returns the same object that has been given to the function, with the given arguments to draw the flowchart stored in the attributes.

## Examples

```
clinic_patient |>  
  as_fc(label = "Available patients") |>  
  fc_filter(age >= 18 & consent == "Yes", label = "Patients included", show_exc = TRUE) |>  
  fc_split(group) |>  
  fc_filter(n_visits == 2, label = "Two visits available", show_exc = TRUE) |>  
  fc_split(marker_alt, label = c("Marker not altered", "Marker altered")) |>  
  fc_draw()
```

---

 fc\_export

*fc\_export*


---

### Description

This function allows you to export the drawn flowchart to the most popular image formats (png, jpeg, tiff).

### Usage

```
fc_export(
  object,
  filename,
  path = NULL,
  format = NULL,
  width = NA,
  height = NA,
  units = "px",
  res = 100
)
```

### Arguments

object	fc object that we want to export.
filename	File name to create on disk.
path	Path of the directory to save plot to: path and filename are combined to create the fully qualified file name. Defaults to the working directory.
format	Format to export the image. One of 'png', 'jpeg' or 'tiff'. If NULL (default), the format is guessed based on the filename extension.
width, height	Plot size in units expressed by the 'units' argument. Default is 600px.
units	One of the following units in which the width and height arguments are expressed: "in", "cm", "mm" or "px". Default is "px".
res	The nominal resolution in ppi which will be recorded in the bitmap file, if a positive integer. Also used for units other than the default, and to convert points to pixels. Default is 100.

### Value

Invisibly returns the same object that has been given to the function.

### Examples

```
## Not run:
clinic_patient |>
  as_fc(label = "Available patients") |>
  fc_filter(age >= 18 & consent == "Yes", label = "Patients included", show_exc = TRUE) |>
```

```

fc_split(group) |>
fc_filter(n_visits == 2, label = "Two visits available", show_exc = TRUE) |>
fc_split(marker_alt, label = c("Marker not altered", "Marker altered")) |>
fc_draw() |>
fc_export("flowchart.png")

```

```
## End(Not run)
```

---

fc\_filter

*fc\_filter*


---

## Description

This function allows to filter the flowchart in function of a expression that returns a logic value that are defined in terms of the variables in the database. It will generate one box per group showing the number of rows of the group that matches the condition, and will retain only those rows in the data base.

## Usage

```

fc_filter(
  object,
  filter = NULL,
  N = NULL,
  label = NULL,
  text_pattern = "{label}\n {n} ({perc}%)",
  show_exc = FALSE,
  direction_exc = "right",
  label_exc = "Excluded",
  text_pattern_exc = "{label}\n {n} ({perc}%)",
  sel_group = NULL,
  round_digits = 2,
  just = "center",
  text_color = "black",
  text_fs = 8,
  bg_fill = "white",
  border_color = "black",
  just_exc = "center",
  text_color_exc = "black",
  text_fs_exc = 6,
  bg_fill_exc = "white",
  border_color_exc = "black"
)

```

**Arguments**

object	fc object that we want to filter.
filter	Expression that returns a logical value and are defined in terms of the variables in the data frame. The data base will be filtered by this expression, and it will create a box showing the number of rows satisfying this condition.
N	Number of rows after the filter in case 'filter' is NULL.
label	Character that will be the title of the box. By default it will be the evaluated condition.
text_pattern	Structure that will have the text in each of the boxes. It recognizes label, n, N and perc within brackets. For default it is "{label}\n {n} ({perc}%)".
show_exc	Logical value. If TRUE a box showing the number of excluded rows will be added to the flow chart.
direction_exc	One of "left" or "right" indicating if the exclusion box goes into the left direction or in the right direction. By default is "right".
label_exc	Character that will be the title of the added box showing the excluded patients. By default it will show "Excluded".
text_pattern_exc	Structure that will have the text in each of the excluded boxes. It recognizes label, n, N and perc within brackets. For default it is "{label}\n {n} ({perc}%)".
sel_group	Specify if the filtering has to be done only by one of the previous groups. By default is NULL.
round_digits	Number of digits to round percentages. It is 2 by default.
just	Justification for the text: left, center or right. Default is center.
text_color	Color of the text. It is black by default.
text_fs	Font size of the text. It is 8 by default.
bg_fill	Box background color. It is white by default.
border_color	Box border color. It is black by default.
just_exc	Justification for the text of the exclude box: left, center or right. Default is center.
text_color_exc	Color of the text of the exclude box. It is black by default.
text_fs_exc	Font size of the text of the exclude box. It is 6 by default.
bg_fill_exc	Exclude box background color. It is white by default.
border_color_exc	Box background color of the exclude box. It is black by default.

**Value**

List with the filtered dataset and the flowchart parameters with the resulting filtered box.



## Examples

```
clinic_patient |>
  as_fc(label = "Patients included") |>
  fc_filter(age >= 18 & consent == "Yes", label = "Patients included", show_exc = TRUE) |>
  fc_draw()
```

---

fc\_merge

*fc\_merge*

---

## Description

This function allows to combine horizontally two different flowcharts.

## Usage

```
fc_merge(fcs)
```

## Arguments

fcs                    list with all the flowcharts that we want to merge

## Value

List containing a list with the datasets belonging to each flowchart and another list with each of the flowcharts parameters to merge.

## Examples

```
#Create first flowchart for patients
fc1 <- clinic_patient |>
  dplyr::filter(!is.na(group)) |>
  as_fc(label = "Patients included") |>
  fc_split(group)

#Create second flowchart for visits
fc2 <- clinic_visit |>
  dplyr::filter(!is.na(group)) |>
  as_fc(label = "Number of visits") |>
  fc_split(group)

list(fc1, fc2) |>
  fc_merge() |>
  fc_draw()
```

fc\_modify

*fc\_modify***Description**

This function allows to modify the ‘\$.fc’ tibble included in each fc object that contains all the parameters of the flowchart.

**Usage**

```
fc_modify(object, fun, ...)
```

**Arguments**

object	flowchart created as a fc object.
fun	A function or formula that will be applied to ‘\$.fc’. If a <code>_function_</code> , it is used as is. If a <code>_formula_</code> , e.g. <code>fun = ~.x  &gt; mutate(x = x + 0.2)</code> , it is converted to a function.
...	Additional arguments passed on to the mapped function.

**Value**

List with the dataset and the modified flowchart parameters.

**Examples**

```
#Example 1 (change text):
clinic_patient |>
  as_fc(label = "Patients included") |>
  fc_filter(age >= 18 & consent == "Yes", label = "Patients included", show_exc = TRUE) |>
  fc_modify(~.x |>
    dplyr::mutate(
      text = dplyr::case_when(
        id == 3 ~ stringr::str_glue("Excluded patients:
          - {sum(clinic_patient$age < 18)} under-age
          - {sum(clinic_patient$consent == 'No')} no consent
        "),
        TRUE ~ text
      )
    ) |>
  fc_draw()
```

```
#Example 2 (change coordinates):
clinic_patient |>
  as_fc(label = "Patients included") |>
  fc_filter(age >= 18 & consent == "Yes", label = "Patients included", show_exc = TRUE) |>
  fc_modify(~.x |>
    dplyr::mutate(
      x = dplyr::case_when(
```

```

        id == 3 ~ 0.8,
        TRUE ~ x
    ),
    y = dplyr::case_when(
        id == 1 ~ 0.85,
        id == 2 ~ 0.15,
        id == 3 ~ 0.5
    )
  )) |>
fc_draw()

```

---

fc\_split

*fc\_split*


---

### Description

This function allows to split the flowchart in function of the categories of a column of the database. It will generate as many boxes as categories has the column showing in each one the frequency of each category. It will additionally group the database per this column.

### Usage

```

fc_split(
  object,
  var = NULL,
  N = NULL,
  label = NULL,
  text_pattern = "{label}\n {n} ({perc}%)",
  sel_group = NULL,
  na.rm = FALSE,
  show_zero = FALSE,
  round_digits = 2,
  just = "center",
  text_color = "black",
  text_fs = 8,
  bg_fill = "white",
  border_color = "black"
)

```

### Arguments

object	fc object that we want to split.
var	variable column of the database from which it will be splitted.
N	Number of rows after the split in case 'var' is NULL.
label	Vector of characters with the label of each category in order. It has to have as many elements as categories has the column. By default, it will put the labels of the categories.

text_pattern	Structure that will have the text in each of the boxes. It recognizes label, n, N and perc within brackets. For default it is "{label}\n {n} ({perc}%)".
sel_group	Specify if the splitting has to be done only by one of the previous groups. By default is NULL.
na.rm	logical. Should missing values of the grouping variable be removed? Default is FALSE.
show_zero	logical. Should the levels of the grouping variable that don't have an event be shown? Default is FALSE.
round_digits	Number of digits to round percentages. It is 2 by default.
just	Justification for the text: left, center or right. Default is center.
text_color	Color of the text. It is black by default.
text_fs	Font size of the text. It is 8 by default.
bg_fill	Box background color. It is white by default.
border_color	Box border color. It is black by default.

**Value**

List with the dataset grouped by the splitting variable and the flowchart parameters with the resulting split.

**Examples**

```
clinic_patient |>
  dplyr::filter(!is.na(group)) |>
  as_fc(label = "Patients included") |>
  fc_split(group) |>
  fc_draw()
```

---

 fc\_stack

*fc\_stack*


---

**Description**

This function allows to combine vertically two different flowcharts.

**Usage**

```
fc_stack(fcs, unite = TRUE)
```

**Arguments**

fcs	list with all the flowcharts that we want to merge
unite	logical value if the boxes have to be united or not

**Value**

List containing a list with the datasets belonging to each flowchart and the flowchart parameters combining all the flowcharts.

**Examples**

```
#Create first flowchart for patients
fc1 <- clinic_patient |>
  dplyr::filter(!is.na(group)) |>
  as_fc(label = "Patients included") |>
  fc_split(group)

#Create second flowchart for visits
fc2 <- clinic_visit |>
  dplyr::filter(!is.na(group)) |>
  as_fc(hide = TRUE) |>
  fc_split(group, label = rep("Number visits", 2), text_pattern = "{label}\\n #' {n}")

list(fc1, fc2) |>
  fc_stack() |>
  fc_draw()
```

---

safo

*Random generated dataset from the SAFO study*

---

**Description**

This dataset is a random generated dataset to reproduce the numbers needed to generate the flowchart of the SAFO study. SAFO is an open-label, multicenter, phase III–IV superiority randomized clinical trial to assess whether cloxacillin plus fosfomycin administered for the initial 7-days of therapy achieves better treatment success than cloxacillin alone in hospitalized patients with MSSA bacteremia.

**Usage**

```
data(safo)
```

**Format**

A data frame with 925 rows and 21 columns

**id:** Identifier of each patient. This information does not match the real data.

**inclusion\_crit:** The patient not met the inclusion criteria?

**exclusion\_crit:** The patient met the exclusion criteria?

**chronic\_heart\_failure:** Exc1: Chronic heart failure?

**expected\_death\_24h:** Exc2: Clinical status with expected death in <24h?

**polymicrobial\_bacteremia:** Exc3: Polymicrobial bacteremia?  
**conditions\_affect\_adherence:** Exc4: Conditions expected to affect adherence to the protocol?  
**susp\_prosthetic\_valve\_endocard:** Exc5: Suspicion of prosthetic valve endocarditis?  
**severe\_liver\_cirrhosis:** Exc6: Severe liver cirrhosis?  
**acute\_sars\_cov2:** Exc7: Acute SARS-CoV-2 infection?  
**blactam\_fosfomicin\_hypersens:** Exc8: Beta-lactam or fosfomicin hypersensitivity?  
**other\_clinical\_trial:** Exc9: Participation in another clinical trial?  
**pregnancy\_or\_breastfeeding:** Exc10: Pregnancy or breastfeeding?  
**previous\_participation:** Exc11: Previous participation in the SAFO trial?  
**myasthenia\_gravis:** Exc12: Myasthenia gravis?  
**decline\_part:** The patient declined to participate?  
**group:** Randomized treatment received: cloxacilin alone / cloxacilin plus fosfomicin  
**itt:** The patient belongs to the intention to treat (ITT) group?  
**reason\_itt:** Reason for exclusion from the ITT group.  
**pp:** The patient belongs to the per protocol (PP) group?  
**reason\_pp:** Reason for exclusion from the PP group.

## References

Grillo, S., Pujol, M., Miró, J.M. et al. Cloxacillin plus fosfomicin versus cloxacillin alone for methicillin-susceptible *Staphylococcus aureus* bacteremia: a randomized trial. *Nat Med* 29, 2518–2525 (2023). <https://doi.org/10.1038/s41591-023-02569-0>

# Index

## \* datasets

- clinic\_patient, 3
- clinic\_visit, 4
- safo, 13

as\_fc, 2

clinic\_patient, 3  
clinic\_visit, 4

fc\_draw, 5  
fc\_export, 6  
fc\_filter, 7  
fc\_merge, 9  
fc\_modify, 10  
fc\_split, 11  
fc\_stack, 12

safo, 13