Package: ggpsychro (via r-universe)

October 12, 2024

Title A 'ggplot2' Extension for Making Pyschrometric Charts

Version 0.0.0.9000

Description A 'ggplot2' extension for making pyschrometric charts.

Depends R (>= 3.2), ggplot2 (>= 3.0.0)

Imports checkmate, psychrolib, scales (>= 1.1.0)

Suggests testthat, covr

License MIT + file LICENSE

URL https://github.com/hongyuanjia/ggpsychro

BugReports https://github.com/hongyuanjia/ggpsychro/issues

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Roxygen list(markdown = TRUE)

Collate 'ggpsychro-global.R' 'aaa-.R' 'geom-grid.R' 'geom-maskarea.R' 'ggproto-classes.R' 'ggpsychro-package.R' 'ggpsychro.R' 'label.R' 'layer.R' 'plot-build.R' 'plot-construction.R' 'psychro-extra.R' 'utils.R' 'trans.R' 'scale.R' 'stat.R' 'theme.R'

Repository https://hongyuanjia.r-universe.dev

RemoteUrl https://github.com/hongyuanjia/ggpsychro

RemoteRef HEAD

RemoteSha 961d96d8b2239556ead600a304aab8b92e61a8a6

Contents

`ggpsychro-package` *ggpsychro: A 'ggplot2' Extension for Making Pyschrometric Charts*

Description

A 'ggplot2' extension for making pyschrometric charts.

Author(s)

Maintainer: Hongyuan Jia <hongyuan.jia@bears-berkeley.sg> (ORCID)

See Also

Useful links:

- https://github.com/hongyuanjia/ggpsychro
- Report bugs at https://github.com/hongyuanjia/ggpsychro/issues

---

`drybulb_trans` *Create transformation objects for psychrometric chart*

Description

Create transformation objects for psychrometric chart

Usage

```
drybulb_trans(units)

humratio_trans(units)

relhum_trans(units)

wetbulb_trans(units)

vappres_trans(units)

specvol_trans(units)

enthalpy_trans(units)
```

Arguments

units	A string indicating the system of units chosen. Should be either "SI" or "IP".

Examples

```
plot(drybulb_trans("SI"), xlim = c(0, 5))
plot(humratio_trans("SI"), xlim = c(0, 1000))
plot(relhum_trans("SI"), xlim = c(0, 1))
plot(wetbulb_trans("SI"), xlim = c(-50, 40))
plot(vappres_trans("SI"), xlim = c(1000, 4000))
plot(specvol_trans("SI"), xlim = c(0.8, 1))
```

---

geom_grid_relhum *Draw grid lines of constant psychrometric properties*

Description

geom_grid_*() geoms draw grid line of constant psychrometric properties, including relative humidity, wet-bulb temperature, water vapor pressure, specific volume and enthalpy, based on current psychrometric chart's dry-bulb temperature range and humidity ratio range.

Usage

```
geom_grid_relhum(
  mapping = NULL,
  data = NULL,
  n = 201,
  label_loc = 0.95,
  label_parse = FALSE,
  units = waiver(),
  pres = waiver(),
  ...,
  na.rm = FALSE
)

geom_grid_wetbulb(
  mapping = NULL,
  data = NULL,
  label_loc = 0.1,
  label_parse = TRUE,
  units = waiver(),
  pres = waiver(),
  ...,
  na.rm = FALSE
)

geom_grid_vappres(
```

```
  mapping = NULL,
  data = NULL,
  label_loc = 0.5,
  label_parse = FALSE,
  units = waiver(),
  pres = waiver(),
  ...,
  na.rm = FALSE
)

geom_grid_specvol(
  mapping = NULL,
  data = NULL,
  label_loc = 0.95,
  label_parse = TRUE,
  units = waiver(),
  pres = waiver(),
  ...,
  na.rm = FALSE
)

geom_grid_enthalpy(
  mapping = NULL,
  data = NULL,
  label_loc = 0.95,
  label_parse = TRUE,
  units = waiver(),
  pres = waiver(),
  ...,
  na.rm = FALSE
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
n	Number of points to interpolate along. Only used in geom_grid_relhum().

`label_loc` A single number in range (0, 1) indicating label position relative to the line. The default values aim to reduce overlappings among differnent psychrometric property lines, but you can change them if needed:

- `0.95` for constant relative humidity lines
- `0.10` for constant wet-bulb temperature lines
- `0.50` for constant water vapor pressure lines
- `0.95` for constant specific volume lines
- `0.95` for constant enthalpy lines

`label_parse` If `TRUE`, the labels will be parsed into expressions and displayed as described in `?plotmath`.

`units` A single string indicating the units sytem to use. Should be either `"SI"` or `"IP"` or waiver()which uses the value from the parent plot. Default:waiver()‘

`pres` A single number indicating the atmosphere pressure in Pa [SI] or Psi [IP]. If `waiver()`, the pressure calculated from the parent plot's altitude value will be used. Default: `waiver()`

`...` Other arguments passed on to `layer()`. These are often aesthetics, used to set an aesthetic to a fixed value, like `colour = "red"` or `size = 3`. They may also be parameters to the paired geom/stat.

`na.rm` If `FALSE`, the default, missing values are removed with a warning. If `TRUE`, missing values are silently removed.

Details

- `geom_grid_relhum()` for relative humidity grid in range [0, 100] in %
- `geom_grid_wetbulb()` for wet-bulb temperature grid in degree_F [IP] or degree_C [SI]
- `geom_grid_vappres()` for partial pressure grid of water vapor in Psi [IP] or Pa [SI]
- `geom_grid_specvol()` for specific volume grid in ft3 lb-1 of dry air [IP] or in m3 kg-1 of dry air [SI]
- `geom_grid_enthalpy()` for moist air enthalpy grid in Btu lb-1 [IP] or kJ kg-1

Each `geom_grid_*()` comes along with a corresponding `scale_*()` function for customizing scale properties, including breaks, labels and etc.

For each psychrometric properties, the maximum and minimum value is calculated based on the ranges of dry-bulb temperature and humidity ratio in the coordinate system.

Alignment

You can modify text alignment with the `vjust` and `hjust` aesthetics. These can either be a number between 0 (right/bottom) and 1 (top/left) or a character (`"left"`, `"middle"`, `"right"`, `"bottom"`, `"center"`, `"top"`). There are two special alignments: `"inward"` and `"outward"`. Inward always aligns text towards the center, and outward aligns it away from the center.

Aesthetics

geom_grid_*() understands the following aesthetics.

- color
- size
- linetype
- alpha
- label.colour
- label.size
- label.angle
- label.hjust
- label.vjust
- label.alpha
- label.family
- label.fontface
- label.lineheight

Examples

```
# add all grid components
ggpsychro() +
    geom_grid_relhum() +
    geom_grid_wetbulb() +
    geom_grid_vappres() +
    geom_grid_specvol() +
    geom_grid_enthalpy()

# custom grid style
ggpsychro() +
   geom_grid_relhum(alpha = 1.0, label.alpha = 1.0, label.size = 6, label.fontface = 2) +
    geom_grid_wetbulb(size = 1.0, color = "black", alpha = 1.0, label_loc = NA) +
    geom_grid_vappres(label.size = 5)
```

---

geom_line_sat *Draw saturation line*

Description

geom_line_sat() draws a saturation line based on current psychrometric chart's dry-bulb temperature (x axis) range and humidity ratio (y axis) range.

Usage

```
geom_line_sat(
  mapping = NULL,
  data = NULL,
  units = waiver(),
  pres = waiver(),
  n = 201,
  ...,
  na.rm = FALSE
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
units	A single string indicating the units sytem to use. Should be either "SI" or "IP" or waiver()which uses the value from the parent plot. Default:waiver()‘
pres	A single number indicating the atmosphere pressure in Pa [SI] or Psi [IP]. If waiver(), the pressure calculated from the parent plot's altitude value will be used. Default: waiver()
n	Number of points to interpolate along
...	Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.

Details

geom_line_sat() is based on ggplot2::geom_line(), so you can further customize the line style in the same way.

Normally there is no need to add another saturation line since ggpsychro() calls geom_line_sat() internally and makes sure that it is always rendered at the last.

Aesthetics

geom_line_sat() is drawing using ggplot2::geom_line() so support the same aesthetics: alpha, color, linetype and size. It also has aesthetics that control the calculation of the saturation line points (required aesthetics are in bold):

See Also

ggpsychro()

Examples

```
# by default, a saturation line is automatically added when calling 'ggpsychro()' function
ggpsychro()

# you can add another saturation line
ggpsychro() + geom_line_sat(units = "SI", pres = 101325, color = "blue", size = 2)
```

---

geom_maskarea *Mask all area outside of saturation line*

Description

geom_maskarea() draws a polygon to mask all area outside of the saturation line. The vertices of polygon are based on current psychrometric chart's dry-bulb temperature (x axis) range and humidity ratio (y axis) range.

Usage

```
geom_maskarea(
  mapping = NULL,
  data = NULL,
  units = waiver(),
  pres = waiver(),
  n = 201,
  ...,
  na.rm = FALSE
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.

data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
units	A single string indicating the units sytem to use. Should be either "SI" or "IP" or waiver()which uses the value from the parent plot. Default:waiver()‘.
pres	A single number indicating the atmosphere pressure in Pa [SI] or Psi [IP]. If waiver(), the pressure calculated from the parent plot's altitude value will be used. Default: waiver().
n	Number of points to interpolate along
...	Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.

Details

geom_maskarea() is based on ggplot2::geom_polygon(), so you can further customize the area style in the same way.

Normally there is no need to add another mask since ggpsychro() calls geom_maskarea() internally and makes sure that it is always rendered after other layers.

Aesthetics

geom_maskarea() understands the following aesthetics (required aesthetics are in bold).

- color
- size
- linetype

Examples

```
# by default, a mask is automatically added when calling 'ggpsychro()' function
ggpsychro()

# replace with another mask area for pressure at 102000
ggpsychro() +
    geom_maskarea(units = "SI", pres = 102000)

# the line style can be further customized like 'ggplot2::geom_line()'
ggpsychro() +
    geom_maskarea(units = "SI", pres = 101325, color = "blue", fill = "green")
```

ggpsychro *Create a ggpsychro plot*

Description

This function is the equivalent of `ggplot2::ggplot()` in ggplot2. It takes care of setting up the plot object along with creating the layout for the plot based on the graph and the specification passed in. Alternatively a layout can be prepared in advance using `create_layout` and passed as the data argument. See *Details* for a description of all available layouts.

Usage

```
ggpsychro(
  data = NULL,
  mapping = aes(),
  tdb_lim = c(0, 50),
  hum_lim = c(0, 50),
  altitude = 0L,
  mask_style = waiver(),
  sat_style = waiver(),
  units = "SI",
  mollier = FALSE
)
```

Arguments

`data`	Default dataset to use for plot. If not already a data.frame, will be converted to one by `ggplot2::fortify()`. If not specified, must be supplied in each layer added to the plot.
`mapping`	Default list of aesthetic mappings to use for plot. If not specified, must be supplied in each layer added to the plot.
`tdb_lim`	A numeric vector of length-2 indicating the dry-bulb temperature limits. Should be in range `[-50, 100]` degree_C [SI] or `[-58, 212]` degree_F [IP]. If `waiver()`, default values will be `[0, 50]` degree_C [SI] or `[30, 120]` degree_F [IP]. Default: `waiver()`.
`hum_lim`	A numeric vector of length-2 indicating the humidity ratio limits. Should be in range `[0, 60]` g_H20 kg_Air-1 [SI] or `[0, 350]` gr_H20 lb_Air-1 [IP]. If `waiver()`, default values will be `[0, 30]` g_H20 kg_Air-1 [SI] or `[0, 210]` gr_H20 lb_Air-1 [IP]. Default: `waiver()`.
`altitude`	A single number of altitude in m [SI] or ft [IP].
`mask_style`	A list containg settings to format mask area. Will be directly passed to `ggplot2::geom_polygon()`. If `waiver()`, defaults below will be used: • `fill`: `white` • `linetype`: `1` • `size`: `0.5`

sat_style	A list containg settings to format saturation line. Will be directly passed to ggplot2::geom_line(). If waiver(), defaults below will be used: • color: #DA251D • linetype: 1 • size: 1 Learn more about setting these aesthetics in vignette("ggplot2-specs").
units	A string indicating the system of units chosen. Should be either "SI" or "IP".
mollier	If TRUE, a Mollier chart will be created instead of a psychrometric chart. Default: FALSE.

Value

An object of class gg onto which layers, scales, etc. can be added.

Author(s)

Hongyuan Jia

Examples

```
ggpsychro()
```

---

label_drybulb *Label wet-bulb temperature*

Description

Format numbers as main variables on the psychrometric chart.

Usage

```
label_drybulb(
  x,
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

label_humratio(
  x,
```

```
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

label_relhum(
  x,
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

label_wetbulb(
  x,
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

label_vappres(
  x,
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

label_specvol(
  x,
```

```
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

label_enthalpy(
  x,
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

drybulb_format(
  x,
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

humratio_format(
  x,
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

relhum_format(
  x,
```

```
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

wetbulb_format(
  x,
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

vappres_format(
  x,
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

specvol_format(
  x,
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)

enthalpy_format(
  x,
```

```
  accuracy = NULL,
  scale = 1,
  units,
  big.mark = ",",
  decimal.mark = ".",
  trim = TRUE,
  parse = FALSE,
  ...
)
```

Arguments

x	A numeric vector
accuracy	A number to round to. Use (e.g.) 0.01 to show 2 decimal places of precision. If NULL, the default, uses a heuristic that should ensure breaks have the minimum number of digits needed to show the difference between adjacent values. Applied to rescaled data.
scale	A scaling factor: x will be multiplied by scale before formating. This is useful if the underlying data is very small or very large.
units	A single string indcating the unit system to use. Should be either "SI" or "IP"
big.mark	Character used between every 3 digits to separate thousands.
decimal.mark	The character to be used to indicate the numeric decimal point.
trim	Logical, if FALSE, values are right-justified to a common width (see base::format()).
parse	If TRUE, the labels will be parsed into expressions and displayed as described in ?plotmath. Default: FALSE.
...	Other arguments passed on to base::format().

Value

All label_() functions return a "labelling" function, i.e. a function that takes a vector x and returns a character vector of length(x) giving a label for each input value.

Labelling functions are designed to be used with the labels argument of ggplot2 scales. The examples demonstrate their use with x scales, but they work similarly for all scales, including those that generate legends rather than axes.

Examples

```
demo_scale(10:50, labels = label_drybulb(units = "SI", parse = TRUE))
demo_scale(10:50, labels = label_drybulb(units = "IP", parse = TRUE))

demo_scale(10:20, labels = label_humratio(scale = 0.001, units = "SI", parse = TRUE))
demo_scale(10:20, labels = label_humratio(scale = 0.007, units = "IP", parse = TRUE))

demo_scale(10:50, labels = label_relhum(units = "SI"))
demo_scale(10:50, labels = label_relhum(units = "IP"))

demo_scale(10:50, labels = label_wetbulb(units = "SI", parse = TRUE))
```

```
demo_scale(10:50, labels = label_wetbulb(units = "IP", parse = TRUE))

demo_scale(10:50, labels = label_specvol(units = "SI", parse = TRUE))
demo_scale(10:50, labels = label_specvol(units = "IP", parse = TRUE))

demo_scale(10:50, labels = label_vappres(units = "SI"))
demo_scale(10:50, labels = label_vappres(units = "IP"))

demo_scale(seq(1000, 2000), labels = label_enthalpy(units = "SI", parse = TRUE))
demo_scale(seq(1000, 2000), labels = label_enthalpy(units = "IP", parse = TRUE))
```

`scale_drybulb_continuous` *Transformation object for psychrometric chart*

Description

Transformation object for psychrometric chart

Usage

```
scale_drybulb_continuous(
  name = waiver(),
  breaks = waiver(),
  minor_breaks = waiver(),
  labels = waiver(),
  limits = NULL,
  units = waiver(),
  ...
)

scale_humratio_continuous(
  name = waiver(),
  breaks = waiver(),
  minor_breaks = waiver(),
  labels = waiver(),
  limits = NULL,
  units = waiver(),
  ...
)

scale_relhum(
  breaks = waiver(),
  minor_breaks = waiver(),
  labels = waiver(),
  units = waiver(),
  ...
```

```
)

scale_wetbulb(
  breaks = waiver(),
  minor_breaks = waiver(),
  labels = waiver(),
  limits = NULL,
  units = waiver(),
  ...
)

scale_vappres(
  breaks = waiver(),
  minor_breaks = waiver(),
  labels = waiver(),
  limits = NULL,
  units = waiver(),
  ...
)

scale_specvol(
  breaks = waiver(),
  minor_breaks = waiver(),
  labels = waiver(),
  limits = NULL,
  units = waiver(),
  ...
)

scale_enthalpy(
  breaks = waiver(),
  minor_breaks = waiver(),
  labels = waiver(),
  limits = NULL,
  units = waiver(),
  ...
)
```

Arguments

`name`	The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted.
`breaks`	One of: • `NULL` for no breaks • `waiver()` for the default breaks computed by the transformation object • A numeric vector of positions

- A function that takes the limits as input and returns breaks as output (e.g., a function returned by `scales::extended_breaks()`)

`minor_breaks` One of:

- `NULL` for no minor breaks
- `waiver()` for the default breaks (one minor break between each major break)
- A numeric vector of positions
- A function that given the limits returns a vector of minor breaks.

`labels` One of:

- `NULL` for no labels
- `waiver()` for the default labels computed by the transformation object
- A character vector giving labels (must be same length as `breaks`)
- A function that takes the breaks as input and returns labels as output

`limits` One of:

- `NULL` to use the default scale range
- A numeric vector of length two providing limits of the scale. Use `NA` to refer to the existing minimum or maximum
- A function that accepts the existing (automatic) limits and returns new limits Note that setting limits on positional scales will **remove** data outside of the limits. If the purpose is to zoom, use the limit argument in the coordinate system (see `coord_cartesian()`).

`units` A string indicating the system of units chosen. Should be:

- `waiver()` for using the parent plot units set in `ggpsychro()`
- Either `"SI"` or `"IP"`

`...` Other arguments passed on to `scale_(x|y)_continuous()`

Examples

```
ggpsychro() +
    geom_grid_relhum() +
    scale_relhum(minor_breaks = NULL) +
    geom_grid_wetbulb() +
    scale_wetbulb(breaks = seq(25, 30, by = 5), minor_breaks = NULL) +
    geom_grid_vappres() +
    scale_vappres(breaks = seq(6000, 7000, by = 500), limits = c(6000, 7000)) +
    geom_grid_specvol() +
    scale_specvol(labels = NULL)
```

stat_relhum *Calculate psychrometric properties of moist air*

Description

Calculate psychrometric properties of moist air

Usage

```
stat_relhum(
  mapping = NULL,
  data = NULL,
  geom = "point",
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

stat_wetbulb(
  mapping = NULL,
  data = NULL,
  geom = "point",
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

stat_vappres(
  mapping = NULL,
  data = NULL,
  geom = "point",
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

stat_specvol(
  mapping = NULL,
  data = NULL,
  geom = "point",
  position = "identity",
```

```
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

stat_enthalpy(
  mapping = NULL,
  data = NULL,
  geom = "point",
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

Arguments

mapping	Set of aesthetic mappings created by aes() or aes_(). If specified and inherit.aes = TRUE (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If NULL, the default, the data is inherited from the plot data as specified in the call to ggplot(). A data.frame, or other object, will override the plot data. All objects will be fortified to produce a data frame. See fortify() for which variables will be created. A function will be called with a single argument, the plot data. The return value must be a data.frame, and will be used as the layer data. A function can be created from a formula (e.g. ~ head(.x, 10)).
geom	The geometric object to use display the data
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.
...	Other arguments passed on to layer(). These are often aesthetics, used to set an aesthetic to a fixed value, like colour = "red" or size = 3. They may also be parameters to the paired geom/stat.
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. borders().

Details

- `stat_relhum` requires an extra `relhum` aesthetics for relative humidity in range [0, 100] in %
- `stat_wetbulb` requires an extra `wetbulb` aesthetics for wet-bulb temperature in degree_F [IP] or degree_C [SI]
- `stat_vappres` requires an extra `vappres` aesthetics for partial pressure of water vapor in moist air in Psi [IP] or Pa [SI]
- `stat_specvol` requires an extra `specvol` aesthetics for specific volume of moist air in ft3 lb-1 of dry air [IP] or in m3 kg-1 of dry air [SI]
- `stat_enthalpy` requires an extra `enthalpy` aesthetics for moist air enthalpy in Btu lb-1 [IP] or J kg-1

What these `ggplot2::ggproto()` objects do are to take input values, calculate the corresponding humidity ratio and replace the `y` aesthetic values in each group.

All of stats above requires two additional aesthetics:

- `units`: A single string indicating the units sytem to use. Should be either `"SI"` or `"IP"` or waiver()`which uses the val`
- `pres`: A single number indicating the atmosphere pressure in Pa [SI] or Psi [IP]. If `waiver()`, the pressure calculated from the parent plot's altitude value will be used. Default: `waiver()`

However, when these stats are used inside a ggplot `geom_*` as the `stat` argument, both `units` and `pres` have to be specified.

Examples

```
p <- ggpsychro() + geom_grid_relhum() # add relative humidity grid lines

# draw a point with dry-bulb at 30C and relative humidity at 60%
p + geom_point(aes(x = 30, relhum = 0.6), stat = "relhum", size = 5)

# draw a constant relative humidity line of 60% with dry-bulb from 20C to 30C
## use stat_* directly
p + stat_relhum(geom = "line", aes(x = 20:30, relhum = 0.6), size = 2)
## OR
## use as thet `stat` argument inside an ggplot `geom_*` function
p + geom_line(aes(x = 20:30, relhum = 0.6), stat = "relhum")
```

Index