Package 'ggcleveland'

August 16, 2021

Title Implementation of Plots from Cleveland's Visualizing Data Book

Version 0.1.0

Description William S. Cleveland's book 'Visualizing Data' is a classic piece of literature on Exploratory Data Analysis. Although it was written several decades ago, its content is still relevant as it proposes several tools which are useful to discover patterns and relationships among the data under study, and also to assess the goodness of fit o a model. This package provides functions to produce the 'ggplot2' versions of the visualization tools described in this book and is thought to be used in the context of courses on Exploratory Data Analysis.

Depends R (>= 3.6.0)

Imports dplyr, tidyr, ggplot2, rlang, magrittr, graphics, readr, egg, vctrs, lattice, tibble, stringr

License GPL-2

Encoding UTF-8

LazyData true

URL https://github.com/mpru/ggcleveland

BugReports https://github.com/mpru/ggcleveland/issues

RoxygenNote 7.1.1

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Marcos Prunello [aut, cre] (<https://orcid.org/0000-0002-9611-527X>), Gonzalo Mari [aut]

Maintainer Marcos Prunello <marcosprunello@gmail.com>

Repository CRAN

Date/Publication 2021-08-16 07:00:09 UTC

R topics documented:

`bin` *Dataset bin*

Description

From Cleveland (1993): Bin packing is a computer problem that has challenged mathematicians working on the foundations of theoretical computer science. Suppose a large number of files of different sizes are to be written on floppies. No file can be split between two floppies, but we want to waste as little space as possible. Unfortunately, any algorithm that guarantees the minimum possible empty space takes an enormous amount of computation time unless the number of files is quite small. Fortunately, there are heuristic algorithms that run fast and do an extremely good job of packing, even though they do not guarantee the minimum of empty space. One is first fit decreasing. The files are packed from largest to smallest. For each file, the first floppy is tried; if it has sufficient empty space, the file is written, and if not, the second floppy is tried. If the second file has sufficient space, the file is written and if not, the third floppy is tried. The algorithm proceeds in this way until a floppy with space, possibly a completely empty one, is found. To supplement the theory of bin packing with empirical results, mathematicians and computer scientists have run simulations, computer experiments in which bins are packed with randomly generated weights. For one data set from one experiment, the weights were randomly selected from the interval 0 to 0.8 and packed in bins of size one. The number of weights, n, for each simulation run took one of

11 values: 125,250,500, and so forth by factors of 2 up to 128000. There were 25 runs for each of the 11 different numbers of weights, which makes 25 x 11 = 275 runs in all. For each run of the experiment, the performance of the algorithm was measured by the total amount of empty space in the bins that were used. We will study log empty space to enhance our understanding of multiplicative effects.

Usage

```
bin
```

Format

A data frame with 275 rows and 2 variables:

empty.space total amount of empty space in the bins that were used

number.weights number of weights

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

`dating` *Dataset dating*

Description

From Cleveland (1993): Ages of many ancient objects are determined by carbon dating. A second dating method, first reported in 1990, provides calibration back to at least 30 kyr BP by measuring the decay of uranium to thorium. The group that invented the method took core samples in coral off the coast of Barbados and dated the material back to nearly 30 kyr BP using both the carbon and thorium methods. The thorium results were used to study the accuracy of the carbon method.

Usage

```
dating
```

Format

A data frame with 19 rows and 2 variables:

carbon carbon age

thorium thorium age

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

environmental *Dataset environmental*

Description

From Cleveland (1993): These measurements were made on 111 days from May to September of 1973 at sites in the New York City metropolitan region; there is one measurement of each variable on each day. Solar radiation is the amount from 0800 to 1200 in the frequency band 4000-7700A, and was measured in Central Park, New York City. Wind speed is the average of values at 0700 and 1000, and was measured at LaGuardia Airport, which is about 7 km from Central Park. Temperature is the daily maximum, and was also measured at LaGuardia. Ozone is the cube root of the average of hourly values from 1300 to 1500, and was measured at Roosevelt Island, which is about 2 km from Central Park and 5 km from LaGuardia.

Usage

```
environmental
```

Format

A data frame with 111 rows and 2 variables:

dia day

ozono ozone

radiacion radiation

temperatura temperature

viento wind

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

equal_count *The equal count algorithm*

Description

This function applies the equal count algorithm to divide a set of observations into intervals which can have certain level of ovelapping. It calls 'lattice::equal.count' but extends the output.

Usage

```
equal_count(df, vble, n_int = 6, frac = 0.5)
```

Arguments

df	dataframe
vble	numeric variable to be analized
n_int	number of intervals
frac	overlapping fraction

Value

a list with two elements:

intervals a tibble where each rows referes to one of the generated interval, with its lower and upper limits, number of values in it and number of values overlapping with the next interval

df_long a tibble in long format where each observation appears as many times as the number of intervals in which it belongs, with an identifier of the observation ('id', its position in the original data.frame) and an identifier of the interval.

Examples

```
equal_count(iris, Sepal.Length, 15, 0.3)
```

---

etanol *Dataset etanol*

Description

From Cleveland (1993): An experiment studied exhaust from an experimental one-cylinder engine fueled by ethanol. The response, which will be denoted by NOx, is the concentration of nitric oxide, NO, plus the concentration of nitrogen dioxide, NO2, normalized by the amount of work of the engine. The units are microg/xg of NOx per joule. One factor is the equivalence ratio, E, at which the engine was run. E is a measure of the richness of the air and fuel mixture; as E increases there is more fuel in the mixture. Another factor is C, the compression ratio to which the engine is set. C is the volume inside the cylinder when the piston is retracted, divided by the volume when the piston is at its maximum point of penetration into the cylinder. There were 88 runs of the experiment.

Usage

```
etanol
```

Format

A data frame with 88 rows and 2 variables:

NOx concentration of nitric oxide plus the concentration of nitrogen dioxide normalized by the amount of work of the engine.

C compression ratio to which the engine is set

E equivalence ratio at which the engine was run

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

`fly` *Dataset fly*

Description

From Cleveland (1993): In 1924, a journal article reported 823 observations from a genetics experiment on flies' eyes. Stocks of the ubiquitous species Drosophila melanogaster Meig were hatched in nine incubators whose temperatures varied from 15°C to 31°C in equal steps of 2°C. The number of facets of the eyes of each hatched fly were reported in units that essentially make the measurement scale logarithmic. The goal of the experiment was to see how facet number depends on temperature.

Usage

```
fly
```

Format

A data frame with 823 rows and 2 variables:

facet number of facets of the eyes

temperature incubator temperature

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

`food` *Dataset food*

Description

From Cleveland (1993): The food web for the animal species in an ecosystem is a description of who eats whom. A chain is a path through the web. It begins with a species that is eaten by no other, moves to a species that the first species eats, moves next to a species that the second species eats, and so forth until the chain ends at a species that preys on no other. If there are 7 species in the chain then there are 6 links between species, and the length of the chain is 6. The mean chain length of a web is the mean of the lengths of all chains in the web. A two-dimensional ecosystem lies in a flat environment such as a lake bottom or a grassland; movement of species in a third dimension is limited. In a three-dimensional ecosystem, there is considerable movement in three dimensions. One example is a forest canopy; another is a water column in an ocean or lake. A mixed ecosystem is made up of a two-dimensional environment and a three-dimensional environment with enough links between the two to regard it as a single ecosystem. An interesting study reports the mean chain lengths for 113 webs.

Usage

```
food
```

Format

A data frame with 113 rows and 2 variables:

mean.length mean web chain length

dimension ecosystem dimenson

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

---

fusion *Dataset fusion*

Description

From Cleveland (1993): An experiment was run to study the effect of prior knowledge of an object's form on fusion time when looking at a stereogram. The experimenters measured the time of first fusion for a particular random dot stereogram. There were two groups of subjects. The NV subjects received either no information or verbal information. The VV subjects received a combination of verbal and visual information, either suggestive drawings of the object or a model of it. Thus the VV subjects actually saw something that depicted the object, but the NV subjects did not. The goal in analyzing the fusion times is to determine if there is a shift in the distribution of the VV times toward lower values compared with the NV times.

Usage

```
fusion
```

Format

A data frame with 78 rows and 2 variables:

time fusion times, seconds

nv.vv experimental group

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

futbol *Dataset futbol*

Description

Data about leg length and kick distance from 300 football players.

Usage

```
futbol
```

Format

A data frame with 300 rows and 2 variables:

longp category of leg length

dist kick distance

Source

Unknown

galaxy *Dataset galaxy*

Description

From Cleveland (1993): NGC 7531 is a spiral galaxy in the Southern Hemisphere. If the only motion of NGC 7531 relative to the earth were the rapid recession due to the big bang, then over the entire region, the velocity relative to the earth would be constant and equal to about 1600 km/sec. But the actual motion is complex. The galaxy appears to be spinning, and there are other motions that are not well understood. The velocity at different points of the galaxy varies by more than 350 km/sec. These data present the locations where 323 measurements were made of the galaxy velocity. The two scales, whose units are arc seconds, are east-west and south-north positions, which form a coordinate system for the celestial sphere based on the earth's standard coordinate system. The goal in analyzing the galaxy data is to determine how the velocity measurements vary over the measurement region; thus velocity is a response and the two coordinate variables are factors.

Usage

```
galaxy
```

Format

A data frame with 323 rows and 6 variables:

ubicacion location number

este.oeste east-west position

norte.sur south-north position

angulo angle

posicion.radial radial position

velocidad velocity

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

`ganglion` *Dataset ganglion*

Description

From Cleveland (1993): For species with highly developed visual systems, such as cats and man, the distribution of ganglion cells across the surface of the retina is not uniform. For example, cats at birth have a much greater density of cells in the central portion of the retina than on the periphery. But in the early stages of fetal development, the distribution of ganglion cells is uniform. The nonuniformity develops in later stages. The data presents the measurement for 14 cat fetuses ranging in age from 35 to 62 days of gestation of the ratio of the central ganglion cell density to the peripheral density and their retinal area, which is nearly monotonically increasing with age.

Usage

```
ganglion
```

Format

A data frame with 14 rows and 2 variables:

area retinal area

cp.ratio ratio of the central ganglion cell density to the peripheral density

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

gg_coplot *Conditional plots*

Description

Implements conditional plots or coplots.

Usage

```
gg_coplot(
  df,
  x,
  y,
  faceting,
  number_bins = 6,
  overlap = 0.5,
  equal_length = TRUE,
  loess = TRUE,
  loess_span = 3/4,
  loess_degree = 1,
  loess_family = "gaussian",
  ylabel = quo_text(y),
  xlabel = quo_text(x),
  facet_label = quo_text(faceting),
  facet_labeller = NULL,
  show_intervals = TRUE,
  intervals_height = 0.25,
  remove_strip = FALSE,
  facets_nrow = NULL,
  hline_at = NULL,
  ...
)
```

Arguments

df	dataframe
x	numeric variable for x-axis
y	numeric variable for y-axis
faceting	faceting numeric variable
number_bins	integer; the number of conditioning intervals
overlap	numeric < 1; the fraction of overlap of the conditioning variables
equal_length	if ‘overlap = 0‘ non overlaping intervals are produced all with same length if ‘equal_length‘ is ‘TRUE‘ (default) or with the same number of values otherwise.
loess	logical; should a loess smoothing curve be added to the coplots? Defaults to TRUE.

loess_span	span parameter for loess
loess_degree	degree parameter for loess
loess_family	famiyly argument for the loess() function
ylabel	label for y-axis
xlabel	label for x-axis
facet_label	label for faceting variable
facet_labeller	defaults to NULL so facet labels are automatically produced, but can take a fuction to be used in 'facet_wrap(~faceting, labeller = labeller(faceting = facet_labeler))'
show_intervals	logical; should the overlapping intervals be shown on their own panel on the top of the figure? Defaults to TRUE.
intervals_height	numeric between 0 and 1, relative size of the intervals pane
remove_strip	logical; should de facets have no strips with labels? Default to FALSE.
facets_nrow	integer; number of rows for the facets
hline_at	numeric; if provide a horizontal line will be added at that heigth
...	addtional parameters passed to geom_point()

Details

If the number of bins is equal to the number of unique values in the faceting variable, then no overlaping intervals are produced and each value in the faceting variable is used as a slice ('frac' is ingored).

If 'overlap = 0' then 'ggplot2::cut_interval' is used to generate the intervals if 'equal_length = TRUE' (default), otherwise 'ggplot2::cut_number' is used. If 'overlap' is not zero, 'graphics::co.interval' is called.

Value

a coplot

Examples

```
data(ruber)

# Slicing con intervalos solapados
gg_coplot(rubber, x = tensile.strength, y = abrasion.loss, faceting = hardness,
  number_bins = 6, overlap = 3/4,
  ylabel = "Pérdida de abrasión (g/hp-hour))",
  xlabel = "Resistencia a la tracción (kg/cm2)",
  facet_label = "Dureza (grados Shore)", loess_family = "symmetric", size = 2)

# Slicing con los valores únicos de la variable de faceting
gg_coplot(galaxy, x = posicion.radial, y = velocidad,
  faceting = angulo, number_bins = 7, loess_span = .5, loess_degree = 2,
  facet_labeller = function(x) paste0("Ángulo = ", x, "º"),
  facet_label = "Ángulo (grado)", facets_nrow = 2, intervals_height = 0.2,
```

```
  xlabel = "Posición radial (arcsec)", ylabel = "Velocidad (km/s)")

data(galaxy)
gg_coplot(galaxy, x = este.oeste, y = norte.sur, faceting = velocidad,
  number_bins = 25, overlap = 0,  size = 0.5,
  ylabel = "Coordenada sur-norte jittered (arcsec)",
  xlabel = "Coordenada este-oeste jittered (arcsec)",
  facet_label = "Velocidad (km/s)", facets_nrow = 5,
  remove_strip = TRUE, intervals_height = 0.15, loess = FALSE)
```

---

gg_pt *Plots for power transformations*

Description

Returns normal QQ plots for a set of power transformations. If there are groups in the data, transformations can be applied separately to each of them.

Usage

```
gg_pt(
  df,
  vble,
  group = NULL,
  taus = c(-1, -0.5, -0.25, 0, 0.25, 0.5, 1),
  xlabel = "Normal quantiles",
  ylabel = paste("Transformed", quo_text(vble)),
  nrow = 2,
  ...
)
```

Arguments

df	dataframe
vble	numeric variable in df to be transformed
group	optional character or factor grouping variable in df. Defaults to NULL.
taus	vector of numeric values for the power transformations (0 is considered to be the log transform)
xlabel	x-axis label
ylabel	y-axis label
nrow	number of rows for facet_wrap, only applied when group is NULL.
...	parameters to be passed to stat_qq(), such as size, color, shape.

Value

a ggplot

Examples

```
library(dplyr)

# Without groups
fusion %>%
  filter(nv.vv == "VV") %>%
  gg_pt(time)

fusion %>%
  filter(nv.vv == "VV") %>%
  gg_pt(time, taus = c(-0.25, -0.5, -1, 0),
        xlabel = "Cuantiles normales", ylabel = "Valores transformados",
        nrow = 3, color = "red")

# With groups
gg_pt(fusion, time, nv.vv, taus = c(-0.5, -0.25, 0, 0.25, 0.5))
```

---

gg_quantiles *Quantile-Quantile plots*

Description

Returns a quantile-quantile plot to compare any given number of groups

Usage

```
gg_quantiles(
  df,
  vble,
  group,
  combined = FALSE,
  xlabel = NULL,
  ylabel = NULL,
  ...
)
```

Arguments

df	dataframe
vble	numeric variable to be analized
group	character or factor grouping variable
combined	logical, defaults to FALSE, producing a matrix of pairwise QQ plots. If TRUE, it produces a QQ plot of quantiles of each group versus quantiles calculated by the combination of all groups. This is useful to study residuals from a fit.
xlabel	label for x-axis
ylabel	label for y-axis
...	parameters to be passed to geom_point(), such as size, color, shape.

Value

a ggplot

Examples

```
library(ggplot2)
data(futbol)

# Multiple groups
gg_quantiles(futbol, dist, longp)
gg_quantiles(futbol, dist, longp, size = 0.4, color = "red", shape = 3) +
  theme(panel.spacing = unit(2, "lines")) +
  theme_bw()

# Only 2 groups
futbol2 <- dplyr::filter(futbol, longp %in% c("< 0.81 m", "0.81 a 0.90 m"))
gg_quantiles(futbol2, dist, longp)

# Each groups vs quantiles from all groups combined
gg_quantiles(futbol, dist, longp, combined = TRUE)
```

---

gg_rf *Residual-Fit plot*

Description

Returns a Residual-Fit plot, optionally including centered observed values

Usage

```
gg_rf(
  df,
  vble,
  fitted,
  res,
  cen_obs = FALSE,
  cen_obs_label = "Centered observed values",
  cen_fit_label = "Centered fitted values",
  res_label = "Residuals",
  xlabel = expression(f[i]),
  ylabel = quo_text(vble),
  ...
)
```

Arguments

df	dataframe
vble	numeric variable in df with the observed values
fitted	numeric variable in df with the fitted values
res	numeric variable in df with the residuals
cen_obs	should centered observed values be included in a panel of their own? Defaults to FALSE. If TRUE, values are centered using the mean of all data
cen_obs_label	label for the panel of centered observed values
cen_fit_label	label for the panel of fitted values
res_label	label for the panel of residuals
xlabel	x-axis label
ylabel	y-axis label
...	parameters to be passed to stat_qq(), such as size, color, shape.

Details

The option to include the centered observed values as part of this plot was inspired by work done by Eng. German Beltzer in lattice.

Value

a ggplot

Examples

```
library(dplyr)
data(futbol)

datos <-
  futbol %>%
  group_by(longp) %>%
  mutate(ajuste = mean(dist), res = dist - ajuste)

gg_rf(datos, dist, ajuste, res)

gg_rf(datos, dist, ajuste, res, cen_obs = TRUE)

gg_rf(datos, dist, ajuste, res, cen_obs = TRUE,
      cen_obs_label = "Obs centradas", cen_fit_label = "Ajustados menos media",
      res_label = "Residuos", xlabel = "valor f", ylabel = "Distancia (m)",
      color = "red", size = 0.7)
```

gg_sl *Spread-Location plot*

Description

Returns a spread-location plot.

Usage

```
gg_sl(
  df,
  vble,
  group,
  jitterwidth = 0.1,
  jitteralpha = 0.5,
  linecol = "red",
  ylabel = expression(sqrt(abs(" Residuals "))),
  xlabel = "Medians"
)
```

Arguments

df	dataframe
vble	numeric variable to be analized
group	grouping character or factor variable
jitterwidth	width argument for geom_jitter
jitteralpha	alpha argument for geom_jitter
linecol	col argument for geom_line
ylabel	y-axis label
xlabel	x-axis label

Value

a ggplot object with the spread-location plot

Examples

```
library(ggplot2)

gg_sl(fusion, time, nv.vv)

gg_sl(fusion, time, nv.vv, jitterwidth = 0.4, linecol = "blue",
      jitteralpha = 1) +
  scale_color_discrete("Grupo") +
  xlim(2, 8)
```

gg_tmd *Tukey's Mean-Difference plot for one-way data*

Description

Returns Tukey's Mean-Difference plot for one-way data

Usage

```
gg_tmd(df, vble, group, xlabel = "Mean", ylabel = "Difference", ...)
```

Arguments

df	dataframe
vble	numeric variable to be analized
group	character or factor grouping variable
xlabel	label for x-axis, defaults to "Mean"
ylabel	label for y-axis, defaults to "Difference"
...	parameters to be passed to geom_point(), such as size, color, shape.

Value

a ggplot

Examples

```
library(dplyr)
data(futbol)

# Multiple groups
gg_tmd(futbol, dist, longp)
gg_tmd(futbol, dist, longp, size = 0.4, color = "red", shape = 3)

# Only 2 groups
futbol %>%
  filter(longp %in% c("< 0.81 m", "0.81 a 0.90 m")) %>%
  gg_tmd(dist, longp)
```

gg_tmd_paired *The gg_tmd_paired function*

Description

Returns Tukey's Mean-Difference plot for paired data (both variables must be measured in the same scale).

Usage

```
gg_tmd_paired(
  df,
  vble1,
  vble2,
  xlabel = "Mean",
  ylabel = "Difference",
  loess = TRUE,
  loess_span = 1,
  loess_degree = 1,
  loess_family = "gaussian",
  ...
)
```

Arguments

df	dataframe
vble1, vble2	numeric variables to be analized
xlabel	label for x-axis, defaults to "Mean"
ylabel	label for y-axis, defaults to "Difference"
loess	logical; should a loess smoothing curve be added to the coplots? Defaults to TRUE.
loess_span	span parameter for loess
loess_degree	degree parameter for loess
loess_family	famiyly argument for the loess() function
...	parameters to be passed to geom_point(), such as size, color, shape.

Details

Differences are computed as 'vble1 - vble2'.

Value

a ggplot

Examples

```
gg_tmd_paired(ozone, stamford, yonkers)
```

`make_coplot_df` *Creation of tibbles por coplots*

Description

It creates dataframes to be used in coplot

Usage

```
make_coplot_df(df, vble, number_bins = 6, overlap = 0.5, equal_length = TRUE)
```

Arguments

`df`	dataframe
`vble`	faceting numeric variable
`number_bins`	integer; the number of conditioning intervals
`overlap`	numeric < 1; the fraction of overlap of the conditioning variables
`equal_length`	if ‘overlap = 0‘ non overlaping intervals are produced all with same length if ‘equal_length‘ is ‘TRUE‘ (default) or with the same number of values otherwise.

Details

Adapted from here.

If ‘overlap = 0‘ then ‘ggplot2::cut_interval‘ is used to generate the intervals if ‘equal_length = TRUE‘ (default), otherwise ‘ggplot2::cut_number‘ is used. If ‘overlap‘ is not zero, ‘graphics::co.interval‘ is called.

Value

a dataset to be used in the creation of coplots

Examples

```
data_coplot <- make_coplot_df(rubber, hardness, 6, 3/4)
```

ozone *Dataset ozone*

Description

From Cleveland (1993): The data are daily maximum ozone concentrations at ground level on 132 days from May 1,1974 to September 30,1974 at two sites in the U.S.A. — Yonkers, New York and Stamford, Connecticut — which are approximately 30 km from one another. The sample for each measurement is the air mass on a particular day, and the bivariate data arise from two measurements at the two sites.

Usage

```
ozone
```

Format

A data frame with 132 rows and 2 variables:

dia day

yonkers air mass at Yonkers

stamford air mass at Stamford

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

playfair *Dataset playfair*

Description

From Cleveland (1993): In 1801, William Playfair published his Statistical Breviary, which contains many displays of economic and demographic data. One display, beautifully reproduced by Tufte, graphs the populations of 22 cities by the areas of circles. The graph also contains a table of the populations, so we can compare the data and the areas of the circles.

Usage

```
playfair
```

Format

A data frame with 22 rows and 2 variables:

city city

population population

diameter diameter of the circle in the figure

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

`polarization` *Dataset polarization*

Description

From Cleveland (1993): This data comes from an experiment on the scattering of sunhght in the atmosphere. One variable is the Babinet point, the scattering angle at which the polarization of sunhght vanishes. The other one is the atmospheric concentration of soHd particles in the air. The goal is to determine the dependence of the Babinet point on concentration.

Usage

```
polarization
```

Format

A data frame with 355 rows and 2 variables:

concentration particulate concentration

babinet Babinet point

Source

Cleveland W. S. (1993). "Visualizing Data". Hobart Press.

`rubber` *Dataset rubber*

Description

From Cleveland (1993): data from an industrial experiment in which thirty rubber specimens were rubbed by an abrasive material. Measurements of three variables - abrasion loss, hardness, and tensile strength - were made for each specimen. Abrasion loss is the amount of material abraded from a specimen per unit of energy expended in the rubbing; tensile strength is the force per unit of cross-sectional area required to break a specimen; and hardness is the rebound height of a steel indenter dropped onto a specimen. The goal is to determine the dependence of abrasion loss on tensile strength and hardness

Usage

```
rubber
```

Format

A data frame with 78 rows and 2 variables:

hardness hardness

tensile.strength tensile strength

abrasion.loss abrasion loss

ts.low tensile.strength - 180 if tensile.strength < 180 or 0 otherwise

ts.high tensile.strength - 180 if tensile.strength > 180 or 0 otherwise

Source

Cleveland W. S. (1993). “Visualizing Data”. Hobart Press.

Index