Package 'liver'

May 8, 2026

Title Toolkit and Datasets for Data Science

Version 1.29

Description Provides a collection of helper functions and illustrative datasets to support learning and teaching of data science with R. The package is designed as a companion to the book <https://book-data-science-r.netlify.app>, making key data science techniques accessible to individuals with minimal coding experience. Functions include tools for data partitioning, performance evaluation, and data transformations (e.g., z-score and min-max scaling). The included datasets are curated to highlight practical applications in data exploration, modeling, and multivariate analysis. An early inspiration for the package came from an ancient Persian idiom about ``eating the liver'', symbolizing deep and immersive engagement with knowledge.

URL https://book-data-science-r.netlify.app

Depends R (>= 3.5.0)

Imports class, ggplot2

Suggests pROC, skimr, knitr, rmarkdown, data.table, mltools, forcats

VignetteBuilder knitr

License GPL (>= 2)

Repository CRAN

Author Reza Mohammadi [aut, cre] (ORCID: <https://orcid.org/0000-0001-9538-0648>), Jeroen van Raak [aut] (ORCID: <https://orcid.org/0000-0002-2190-0126>), Kevin Burke [aut] (ORCID: <https://orcid.org/0000-0001-8724-809X>)

Maintainer Reza Mohammadi <a.mohammadi@uva.nl>

NeedsCompilation no

Date/Publication 2026-05-04 10:10:02 UTC

Contents

`liver-package` *liver: Foundations Toolkit and Datasets for Data Science*

Description

The **liver** package provides a collection of helper functions and illustrative datasets to support learning and teaching of data science with R. The package is designed as a companion to the book Data Science Foundations and Machine Learning Using R, making key data science techniques accessible to individuals with minimal coding experience. Functions include tools for data partitioning, performance evaluation, and data transformations (e.g., z-score and min-max scaling). The included datasets are curated to highlight practical applications in data exploration, modeling, and multivariate analysis. An early inspiration for the package came from an ancient Persian idiom about "eating the liver," symbolizing deep and immersive engagement with knowledge.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>
Amsterdam Business School
University of Amsterdam

Kevin Burke <kevin.burke@ul.ie>
Departement of Statistics
University of Limerick

Maintainer: Reza Mohammadi <a.mohammadi@uva.nl>

`accuracy` *Average classification accuracy*

Description

Computes average classification accuracy.

Usage

```
accuracy(pred, actual, cutoff = NULL, reference = NULL)
```

Arguments

`pred`	a numerical vector of estimated values.
`actual`	a numerical vector of actual values.
`cutoff`	cutoff value for the case that `pred` is vector of probabilites.
`reference`	a factor of classes to be used as the true results.

Value

the computed average classification accuracy (numeric value).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

See Also

conf.mat, mse, mae

Examples

```
pred   = c("no", "yes", "yes", "no", "no", "yes", "no", "no")
actual = c("yes", "no", "yes", "no", "no", "no", "yes", "yes")

accuracy(pred, actual)
```

---

adult *adult data set*

Description

the adult dataset was collected from the US Census Bureau and the primary task is to predict whether a given adult makes more than $50K a year based attributes such as education, hours of work per week, etc. the target feature is *income*, a factor with levels "<=50K" and ">50K", and the remaining 14 variables are predictors.

Usage

```
data(adult)
```

Format

the adult dataset, as a data frame, contains 48598 rows and 15 columns (variables/features). the 15 variables are:

- age: age in years.
- workclass: a factor with 6 levels.
- demogweight: the demographics to describe a person.
- education: a factor with 16 levels.
- education.num: an ordinal encoding of the 'education' feature.
- marital.status: a factor with 5 levels.
- occupation: a factor with 15 levels.
- relationship: a factor with 6 levels.
- race: a factor with 5 levels.

- gender: a factor with levels "Female","Male".
- capital.gain: capital gains.
- capital.loss: capital losses.
- hours.per.week: number of hours of work per week.
- native.country: a factor with 42 levels.
- income: yearly income as a factor with levels "<=50K" and ">50K".

Source

The data are based on the Adult, or Census Income, dataset from the UCI Machine Learning Repository. The original extraction was performed by Barry Becker from the 1994 Census database.

The dataset is also associated with DOI:

doi:10.24432/C5XW20

References

Kohavi, R. and Becker, B. (1996). Adult. UCI Machine Learning Repository. doi:10.24432/C5XW20

Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. *Kdd*.

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, risk, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(adult)
str(adult)
```

---

advertising *advertising data set*

Description

the dataset is from an anonymous organisation's social media ad campaign. the advertising dataset contains 11 features and 1143 records.

Usage

```
data(advertising)
```

Format

the `advertising` dataset, as a data frame, contains 1143 rows and 11 columns (variables/features). the 11 variables are:

- `ad.id`: an unique ID for each ad.
- `xyz.campaign.id`: an ID associated with each ad campaign of XYZ company.
- `fb.campaign.id`: an ID associated with how Facebook tracks each campaign.
- `age`: age of the person to whom the ad is shown.
- `gender`: gender of the person to whim the add is shown.
- `interest`: a code specifying the category to which the person's interest belongs (interests are as mentioned in the person's Facebook public profile).
- `impressions`: the number of times the ad was shown.
- `clicks`: number of clicks on for that ad.
- `spend`: amount paid by company xyz to Facebook, to show that ad.
- `conversion`: total number of people who enquired about the product after seeing the ad.
- `approved`: total number of people who bought the product after seeing the ad.

Details

For more information related to the dataset see:
https://www.kaggle.com/loveall/clicks-conversion-tracking

Source

This dataset is from:
https://www.kaggle.com/loveall/clicks-conversion-tracking

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, risk, cereal, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(advertising)
str(advertising)
```

bank *Bank marketing data set*

Description

the data is related to direct marketing campaigns of a Portuguese banking institution. the marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be (or not) subscribed. the classification goal is to predict if the client will subscribe a term deposit (variable deposit).

Usage

```
data(bank)
```

Format

the bank dataset, as a data frame, contains 4521 rows (customers) and 17 columns (variables/features). the 17 variables are:

Bank client data:

- age: numeric.
- job: type of job; categorical: "admin.", "unknown", "unemployed", "management", "housemaid", "entrepreneur", "student", "blue-collar, "self-employed", "retired", "technician", "services".
- marital: marital status; categorical: "married", "divorced", "single"; note: "divorced" means divorced or widowed.
- education: categorical: "secondary", "primary", "tertiary", "unknown".
- default: has credit in default?; binary: "yes","no".
- balance: average yearly balance, in euros; numeric.
- housing: has housing loan? binary: "yes", "no".
- loan: has personal loan? binary: "yes", "no".

Related with the last contact of the current campaign:

- contact: contact: contact communication type; categorical: "unknown","telephone","cellular".
- day: last contact day of the month; numeric.
- month: last contact month of year; categorical: "jan", "feb", "mar", ..., "nov", "dec".
- duration: last contact duration, in seconds; numeric.

Other attributes:

- campaign: number of contacts performed during this campaign and for this client; numeric, includes last contact.
- pdays: number of days that passed by after the client was last contacted from a previous campaign; numeric, -1 means client was not previously contacted.

- `previous`: number of contacts performed before this campaign and for this client; numeric.
- `poutcome`: outcome of the previous marketing campaign; categorical: "success", "failure", "unknown", "other".

Target variable:

- `deposit`: Indicator of whether the client subscribed a term deposit; binary: "yes" or "no".

Details

For more information related to the dataset see:
http://archive.ics.uci.edu/ml/datasets/Bank+Marketing

Source

This dataset comes from the UCI repository of machine learning databases:
http://archive.ics.uci.edu/ml/datasets/Bank+Marketing

References

Moro, S., Laureano, R. and Cortez, P. (2011) Using Data Mining for Bank Direct Marketing: An Application of the CRISP-DM Methodology. In P. Novais et al. (Eds.), Proceedings of the European Simulation and Modelling Conference.

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

churn_mlc, churn, churn_tel, adult, risk, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(bank)
str(bank)
```

---

bike_demand *Seoul Bike Sharing Demand Data*

Description

A dataset containing hourly bike rental demand in Seoul, South Korea, together with weather conditions, seasonal information, holiday status, and whether the bike sharing system was operating on that day.

Usage

```
data(bike_demand)
```

Format

A data frame with 8760 observations and 14 variables:

date Date of observation.

hour Hour of the day, ranging from 0 to 23.

temperature Temperature in degrees Celsius.

humidity Humidity percentage.

wind_speed Wind speed in meters per second.

visibility Visibility in units recorded by the source dataset.

dew_point_temperature Dew point temperature in degrees Celsius.

solar_radiation Solar radiation in megajoules per square meter.

rainfall Rainfall in millimeters.

snowfall Snowfall in centimeters.

season Season of the year: `"spring"`, `"summer"`, `"autumn"`, or `"winter"`.

holiday Holiday status: `"holiday"` or `"no holiday"`.

functioning_day Whether the bike rental system was operating: `"yes"` or `"no"`.

bike_count Number of rented bikes (target variable).

Details

This dataset was obtained from the UCI Machine Learning Repository and renamed `bike_demand` for inclusion in the **liver** package. It can be used to illustrate methods for regression, exploratory data analysis, and predictive modeling in R.

Source

https://archive.ics.uci.edu/dataset/560/seoul+bike+sharing+demand

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

`mortgage`, `bank`, `churn_mlc`, `churn`, `churn_tel`, `adult`, `cereal`, `advertising`, `marketing`, `drug`, `house`, `house_price`, `red_wines`, `white_wines`, `insurance`, `caravan`, `loan`

Examples

```
data(bike_demand)

str(bike_demand)
summary(bike_demand)
```

caravan *Caravan insurance data set*

Description

The contains 5822 customer records from an insurance company, each described by 86 variables. These include 43 sociodemographic features based on zip codes and 43 indicators of product ownership. The final variable, Purchase, indicates whether a customer bought a caravan insurance policy. Collected for the CoIL 2000 Challenge, the data was designed to address the question: *Can you predict who would be interested in buying a caravan insurance policy and explain why?*

Usage

```
data(caravan)
```

Format

A data frame with 5822 observations (rows) and 86 features (columns).

Details

For more information related to the dataset see
https://www.kaggle.com/datasets/uciml/caravan-insurance-challenge

Source

The data was supplied by Sentient Machine Research: https://www.smr.nl

References

P. van der Putten and M. van Someren (eds) . CoIL Challenge 2000: The Insurance Company Case. Published by Sentient Machine Research, Amsterdam. Also a Leiden Institute of Advanced Computer Science Technical Report 2000-09. June 22, 2000.
James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An Introduction to Statistical Learning with applications in R, https://www.statlearning.com, *Springer-Verlag*.

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, risk, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, loan

Examples

```
data(caravan)
str(caravan)
```

cereal *Cereal data set*

Description

This dataset contains nutrition information for 77 breakfast cereals and includes 16 variables. the "rating" column is our target as a rating of the cereals (Possibly from Consumer Reports?).

Usage

```
data(cereal)
```

Format

the cereal dataset, as a data frame, contains 77 rows (breakfast cereals) and 16 columns (variables/features). the 16 variables are:

- name: Name of cereal.
- manuf: Manufacturer of cereal, coded into seven categories: "A" for American Home Food Products, "G" for General Mills, "K" for Kelloggs, "N" for Nabisco, "P" for Post, "Q" for Quaker Oats, and "R" for Ralston Purina.
- type: cold or hot.
- calories: calories per serving.
- protein: grams of protein.
- fat: grams of fat.
- sodium: milligrams of sodium.
- fiber: grams of dietary fiber.
- carbo: grams of complex carbohydrates.
- sugars: grams of sugars.
- potass: milligrams of potassium.
- vitamins: vitamins and minerals - 0, 25, or 100, indicating the typical percentage of FDA recommended.
- shelf: display shelf (1, 2, or 3, counting from the floor).
- weight: weight in ounces of one serving.
- cups: number of cups in one serving.
- rating: a rating of the cereals (Possibly from Consumer Reports?).

Details

For more information related to the dataset see
https://www.openml.org/search?type=data&status=any&id=1095&sort=runs

Source

This dataset is originally from
https://lib.stat.cmu.edu/DASL/Datafiles/Cereals.html

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, risk, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(cereal)
str(cereal)
```

---

churn *Churn dataset for Credit Card Customers*

Description

The *churn* data set contains 10127 rows (customers) and 21 columns (features). The *churn* column is our target which indicate whether customer churned (left the company) or not.

Usage

```
data(churn)
```

Format

the churn dataset, as a data frame, contains 10127 rows (customers) and 21 columns (variables/features). the 21 variables are:

- `customer.ID`: Unique identifier for each account holder.
- `age`: Age of the customer, in years.
- `gender`: Gender of the account holder.
- `education`: Educational qualification (high-school, college, graduate, uneducated, post-graduate, doctorate, unknown).
- `marital`: Marital status (married, single, divorced, unknown).
- `income`: Annual income bracket (less than $40K, $40K-$60K, $60K-$80K, $80K-$120K, over $120K, unknown).
- `card.category`: Credit card type (blue, silver, gold, platinum).
- `dependent.count`: Number of dependents.

- `months.on.book`: Tenure with the bank, in months.
- `relationship.count`: Total number of products held by the customer (1-6).
- `months.inactive`: Number of inactive months in the past 12 months.
- `contacts.count.12`: Number of customer service contacts in the past 12 months.
- `credit.limit`: Total credit card limit.
- `revolving.balance`: Current revolving balance on the credit card.
- `available.credit`: Available credit line, representing the unused portion of the credit limit. Calculated as `credit.limit - revolving.balance`.
- `transaction.amount.12`: Total transaction amount in the past 12 months.
- `transaction.count.12`: Total number of transactions in the past 12 months.
- `ratio.amount.Q4.Q1`: Ratio of total transaction amount in the fourth quarter to that in the first quarter.
- `ratio.count.Q4.Q1`: Ratio of total transaction count in the fourth quarter to that in the first quarter.
- `utilization.ratio`: Average credit utilization ratio, defined as `revolving.balance / credit.limit`.
- `churn`: Indicator of whether the account was closed (`yes`) or remained active (`no`).

Details

For more information related to the dataset see:
https://www.kaggle.com/sakshigoyal7/credit-card-customers

Source

This dataset is originally from https://leaps.analyttica.com/home

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn_tel, adult, risk, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(churn)

str(churn)
```

churn_mlc *Churn data set from MLC++ machine learing*

Description

This dataset originates from the MLC++ machine learning software and is used for modeling customer churn. Customer *churn*, also known as customer attrition, refers to the event in which customers stop doing business with a company. The dataset contains 5000 rows (customers) and 20 columns (features). The *churn* column serves as the target variable, indicating whether a customer has churned (left the company) or not.

Usage

```
data(churn_mlc)
```

Format

A data frame with 5000 rows (customers) and 20 columns (variables/features). the 20 variables are:

- `state`: Categorical, for the 51 states and the District of Columbia.
- `area.code`: Categorical.
- `account.length`: count, how long account has been active.
- `voice.plan`: Categorical, yes or no, voice mail plan.
- `voice.messages`: Count, number of voice mail messages.
- `intl.plan`: Categorical, yes or no, international plan.
- `intl.mins`: Continuous, minutes customer used service to make international calls.
- `intl.calls`: Count, total number of international calls.
- `intl.charge`: Continuous, total international charge.
- `day.mins`: Continuous, minutes customer used service during the day.
- `day.calls`: Count, total number of calls during the day.
- `day.charge`: Continuous, total charge during the day.
- `eve.mins`: Continuous, minutes customer used service during the evening.
- `eve.calls`: Count, total number of calls during the evening.
- `eve.charge`: Continuous, total charge during the evening.
- `night.mins`: Continuous, minutes customer used service during the night.
- `night.calls`: Count, total number of calls during the night.
- `night.charge`: Continuous, total charge during the night.
- `customer.calls`: Count, number of calls to customer service.
- `churn`: Categorical, yes or no. Indicator of whether the customer has left the company (yes or no).

Details

For more information related to the dataset see
- OpenML: https://www.openml.org/search?type=data&sort=runs&id=40701&status=active
- data.world: https://data.world/earino/churn

Source

This dataset is originally from http://www.sgi.com/tech/mlc

References

Saha, S., Saha, C., Haque, M. M., Alam, M. G. R., and Talukder, A. (2024). ChurnNet: Deep learning enhanced customer churn prediction in telecommunication industry. *IEEE access*, 12, 4471-4484.
Umayaparvathi, V., and Iyakutti, K. (2016). A survey on customer churn prediction in telecom industry: Datasets, methods and metrics. *International Research Journal of Engineering and Technology (IRJET)*, 3(04), 1065-1070
Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn, churn_tel, adult, risk, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(churn_mlc)
str(churn_mlc)
```

---

churn_tel *churn_tel dataset*

Description

The *churn_tel* data set contains 7043 rows (customers) and 21 columns (features). The *churn* column is our target which indicate whether customer churned (left the company) or not.

Usage

```
data(churn_tel)
```

Format

the `churn_tel` dataset, as a data frame, contains 7043 rows (customers) and 21 columns (variables/features). the 21 variables are:

- `customer.ID`: Customer ID.
- `gender`: Whether the customer is a male or a female.
- `senior.citizen`: Whether the customer is a senior citizen or not (1, 0).
- `partner`: Whether the customer has a partner or not (yes, no).
- `dependent`: Whether the customer has dependents or not (yes, no).
- `tenure`: Number of months the customer has stayed with the company.
- `phone.service`: Whether the customer has a phone service or not (yes, no).
- `multiple.lines`: Whether the customer has multiple lines or not (yes, no, no phone service).
- `internet.service`: Customer's internet service provider (DSL, fiber optic, no).
- `online.security`: Whether the customer has online security or not (yes, no, no internet service).
- `online.backup`: Whether the customer has online backup or not (yes, no, no internet service).
- `device.protection`: Whether the customer has device protection or not (yes, no, no internet service).
- `tech.support`: Whether the customer has tech support or not (yes, no, no internet service).
- `streaming.TV`: Whether the customer has streaming TV or not (yes, no, no internet service).
- `streaming.movie`: Whether the customer has streaming movies or not (yes, no, no internet service).
- `contract`: the contract term of the customer (month to month, 1 year, 2 year).
- `paperless.bill`: Whether the customer has paperless billing or not (yes, no).
- `payment.method`: the customer's payment method (electronic check, mail check, bank transfer, credit card).
- `monthly.charge`: the amount charged to the customer monthly.
- `total.charges`: the total amount charged to the customer.
- `churn`: Whether the customer churned or not (yes or no).

Details

For more information related to the dataset see:
https://www.kaggle.com/blastchar/telco-customer-churn

Source

This dataset comes from the IBM Sample Data Sets:
https://community.ibm.com/community/user/blogs/steven-macko/2019/07/11/telco-customer-churn-1113

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, adult, risk, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(churn_tel)
str(churn_tel)
```

`conf.mat` *Confusion Matrix*

Description

Create a Confusion Matrix.

Usage

```
conf.mat(pred, actual, cutoff = 0.5, reference = NULL,
         proportion = FALSE, dnn = c("Actual", "Predict"), ...)
```

Arguments

`pred`	a vector of estimated values.
`actual`	a vector of actual values.
`cutoff`	cutoff value for the case that `pred` is vector of probabilites.
`reference`	a factor of classes to be used as the true results.
`proportion`	Logical: FALSE (default) for a confusion matrix with number of cases. TRUE, for a confusion matrix with the proportion of cases.
`dnn`	the names to be given to the dimensions in the result (the dimnames names).
`...`	options to be passed to `table`.

Value

the results of `table` on `pred` and `actual`.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

See Also

conf.mat.plot, accuracy

Examples

```
pred   = c("no", "yes", "yes", "no", "no", "yes", "no", "no")
actual = c("yes", "no", "yes", "no", "no", "no", "yes", "yes")

conf.mat(pred, actual)
conf.mat(pred, actual, proportion = TRUE)
```

---

conf.mat.plot *Plot Confusion Matrix*

Description

Plot a Confusion Matrix.

Usage

```
conf.mat.plot(pred, actual, cutoff = 0.5, reference = NULL, conf.level = 0,
              margin = c(1, 2), color = c("#F4A582", "#A8D5BA"), ...)
```

Arguments

pred	a vector of estimated values.
actual	a vector of actual values.
cutoff	cutoff value for the case that pred is vector of probabilites.
reference	a factor of classes to be used as the true results.
conf.level	confidence level used for the confidence rings on the odds ratios. Must be a single nonnegative number less than 1; if set to 0 (the default), confidence rings are suppressed.
margin	a numeric vector with the margins to equate. Must be one of 1, 2, or c(1, 2) (the default), which corresponds to standardizing the row, column, or both margins in each 2 by 2 table. Only used if std equals "margins".
color	a vector of length 2 specifying the colors to use for the smaller and larger diagonals of each 2 by 2 table.
...	options to be passed to fourfoldplot.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

See Also

conf.mat

Examples

```
pred   = c("no", "yes", "yes", "no", "no", "yes", "no", "no")
actual = c("yes", "no", "yes", "no", "no", "no", "yes", "yes")

conf.mat.plot(pred, actual)
```

---

cpu_price *CPU Specifications and Market Prices*

Description

A dataset containing detailed specifications, integrated graphics availability, and market price information for a range of computer processors (CPUs). It includes hardware characteristics such as core counts, thread counts, clock speeds, cache size, and thermal design power (TDP), along with price data. The dataset is suitable for studying price-to-performance trade-offs across different CPU models.

Usage

```
data(cpu_price)
```

Format

A data frame with 45 observations and 12 variables:

model The model name of the processor.

brand The brand of the CPU: `"AMD"` or `"Intel"`.

gpu Whether the CPU includes integrated graphics: `"yes"` or `"no"`.

architecture The microarchitecture or generation family of the CPU.

base_ghz The base operating frequency of the CPU in gigahertz.

boost_ghz The maximum turbo or boost frequency of the CPU in gigahertz.

p_cores The number of performance cores (P-cores).

e_cores The number of efficiency cores (E-cores).

threads The number of logical threads the CPU can execute simultaneously.

cache The total cache size in megabytes.

tdp The typical thermal design power (TDP) of the CPU in watts under standard load conditions.

price The approximate retail market price of the CPU in US dollars.

Details

The dataset was assembled to support exploratory and predictive analyses of CPU pricing. For example, it can be used in regression models relating CPU price to processor characteristics such as clock speed, thread count, graphics support, and brand.

Source

The dataset was collected by the package authors. Hardware specifications are based on publicly available manufacturer information. Price data was collected through Google searches during Spring 2026 and reflects approximate retail market prices at that time.

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bike_demand, mortgage, bank, churn_mlc, churn, churn_tel, adult, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(cpu_price)

str(cpu_price)
summary(cpu_price)
```

---

credit *South German Credit Data*

Description

A dataset containing information on credit applicants, including account status, credit history, loan purpose, credit amount, savings, employment duration, personal characteristics, property, housing, and other financial attributes. The outcome variable indicates whether the applicant represents a good or bad credit risk.

Usage

```
data(credit)
```

Format

A data frame with 1000 observations and 21 variables:

status Status of the debtor's checking account with the bank.

duration Credit duration in months.

credit_history History of compliance with previous or concurrent credit contracts.

purpose Purpose for which the credit is needed.

amount Credit amount in Deutsche Mark (DM).

savings Debtor's savings.

employment_duration Duration of the debtor's employment with the current employer.

installment_rate Credit installments as a percentage of the debtor's disposable income.

personal_status_sex Combined information on personal status and sex.

other_debtors Whether there is another debtor or a guarantor for the credit.

present_residence Length of time the debtor has lived in the present residence.

property The debtor's most valuable property.

age Age in years.

other_installment_plans Installment plans from providers other than the credit-giving bank.

housing Type of housing the debtor lives in.

number_credits Number of credits the debtor has or had at this bank, including the current one.

job Quality of the debtor's job.

people_liable Number of persons financially dependent on the debtor.

telephone Whether a telephone landline is registered in the debtor's name.

foreign_worker Whether the debtor is a foreign worker.

credit_risk Credit risk outcome: `"good risk"` or `"bad risk"`.

Details

The South German Credit data are a corrected and documented version of the widely used German credit data. The dataset contains 700 good and 300 bad credits and covers actual credit data from 1973 to 1975, with bad credits heavily oversampled. It can be used to illustrate methods for classification, exploratory data analysis, and predictive modeling in R.

Source

UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/573/south+german+credit+update

South German Credit [Dataset]. (2020). UCI Machine Learning Repository. doi:10.24432/C5QG88

References

Gr\"omping, U. (2019). South German credit data: Correcting a widely used data set.

See Also

`mortgage`, `bank`, `churn_mlc`, `churn`, `churn_tel`, `adult`, `cereal`, `advertising`, `marketing`, `drug`, `house`, `house_price`, `red_wines`, `white_wines`, `insurance`, `caravan`, `loan`

Examples

```
data(credit)

str(credit)
summary(credit)
```

`creditcard_fraud` *Credit Card Transactions for Fraud Detection*

Description

A dataset containing credit card transactions for illustrating fraud detection and class imbalance in binary classification. The data include anonymized predictors derived from a principal component analysis, together with transaction time, transaction amount, and a binary fraud indicator.

Usage

```
data(creditcard_fraud)
```

Format

A data frame with 10000 observations and 31 variables:

Time Seconds elapsed between each transaction and the first transaction in the dataset.

V1 Anonymized predictor obtained from a PCA transformation of the original variables.

V2 Anonymized predictor obtained from a PCA transformation of the original variables.

V3 Anonymized predictor obtained from a PCA transformation of the original variables.

V4 Anonymized predictor obtained from a PCA transformation of the original variables.

V5 Anonymized predictor obtained from a PCA transformation of the original variables.

V6 Anonymized predictor obtained from a PCA transformation of the original variables.

V7 Anonymized predictor obtained from a PCA transformation of the original variables.

V8 Anonymized predictor obtained from a PCA transformation of the original variables.

V9 Anonymized predictor obtained from a PCA transformation of the original variables.

V10 Anonymized predictor obtained from a PCA transformation of the original variables.

V11 Anonymized predictor obtained from a PCA transformation of the original variables.

V12 Anonymized predictor obtained from a PCA transformation of the original variables.

V13 Anonymized predictor obtained from a PCA transformation of the original variables.

V14 Anonymized predictor obtained from a PCA transformation of the original variables.

V15 Anonymized predictor obtained from a PCA transformation of the original variables.

V16 Anonymized predictor obtained from a PCA transformation of the original variables.

V17 Anonymized predictor obtained from a PCA transformation of the original variables.

V18 Anonymized predictor obtained from a PCA transformation of the original variables.

V19 Anonymized predictor obtained from a PCA transformation of the original variables.

V20 Anonymized predictor obtained from a PCA transformation of the original variables.

V21 Anonymized predictor obtained from a PCA transformation of the original variables.

V22 Anonymized predictor obtained from a PCA transformation of the original variables.

V23 Anonymized predictor obtained from a PCA transformation of the original variables.

V24 Anonymized predictor obtained from a PCA transformation of the original variables.

V25 Anonymized predictor obtained from a PCA transformation of the original variables.

V26 Anonymized predictor obtained from a PCA transformation of the original variables.

V27 Anonymized predictor obtained from a PCA transformation of the original variables.

V28 Anonymized predictor obtained from a PCA transformation of the original variables.

Amount Transaction amount.

Class Fraud indicator: `0` for non-fraudulent transactions and `1` for fraudulent transactions.

Details

This dataset is a teaching subset derived from the original Credit Card Fraud Detection dataset available on Kaggle. The original dataset is highly imbalanced. For inclusion in the **liver** package, we created a smaller subset with 10000 observations that retains all fraud cases and a random sample of non-fraud cases. This version is intended for illustrating class imbalance, resampling strategies, and model evaluation in binary classification.

Source

https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

References

Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, and Gianluca Bontempi (2015). Calibrating Probability with Undersampling for Unbalanced Classification. In *2015 IEEE Symposium Series on Computational Intelligence*.

Reza Mohammadi (2025). *Data Science Foundations and Machine Learning with R: From Data to Decisions*. https://book-data-science-r.netlify.app.

See Also

mortgage, bank, churn_mlc, churn, churn_tel, adult, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(creditcard_fraud)
str(creditcard_fraud)

table(creditcard_fraud$Class)
```

`doctor_visits` *Doctor Visits and Health Care Utilization Data*

Description

A dataset containing information on individuals' doctor visit counts, demographic characteristics, income, illness burden, reduced activity days, self-reported health status, and indicators of health care coverage and chronic conditions.

Usage

```
data(doctor_visits)
```

Format

A data frame with 5190 observations and 12 variables:

age Age of the individual.

income Income level of the individual.

illness Number of illnesses experienced by the individual.

reduced Number of days with reduced activity.

health Self-reported health score.

gender Gender of the individual: `"male"` or `"female"`.

private Whether the individual has private health insurance: `"yes"` or `"no"`.

freepoor Whether the individual is covered by free government health care due to low income: `"yes"` or `"no"`.

freerepat Whether the individual is covered by free government health care due to repatriation status: `"yes"` or `"no"`.

nchronic Whether the individual has a chronic condition that is not limiting: `"yes"` or `"no"`.

lchronic Whether the individual has a chronic condition that is limiting: `"yes"` or `"no"`.

visits Number of doctor visits (target variable).

Details

This dataset was adapted for inclusion in the **liver** package and can be used to illustrate methods for count data modeling, exploratory data analysis, and regression techniques such as Poisson regression in R.

Source

Originally distributed with the **AER** package.

References

Mullahy, J. (1997). Heterogeneity, Excess Zeros, and the Structure of Count Data Models. *Journal of Applied Econometrics*, 12:337–350.

Cameron, A.C. and Trivedi, P.K. (1986). Econometric Models Based on Count Data: Comparisons and Applications of Some Estimators and Tests. *Journal of Applied Econometrics*, 1:29–53.

Cameron, A.C. and Trivedi, P.K. (1998). *Regression Analysis of Count Data*. Cambridge: Cambridge University Press.

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bike_demand, mortgage, bank, churn_mlc, churn, churn_tel, adult, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(doctor_visits)

str(doctor_visits)
summary(doctor_visits)
```

---

drug *drug data set*

Description

synthetically generated dataset of 200 patients includes their age, sodium-to-potassium (Na/K) ratio, and the prescribed drug type.

Usage

```
data(drug)
```

Format

the drug dataset, as a data frame, contains 200 rows (customers) and 3 columns (variables/features). the 3 variables are:

- age: age of patients.
- ratio: sodium-to-potassium (Na/K) ratio.
- type: the prescribed drug type in three levels (A, B, and C).

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, risk, cereal, advertising, marketing, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(drug)
str(drug)
```

---

find.na *find.na*

Description

Finding missing values.

Usage

```
find.na(x)
```

Arguments

x	a numerical vector, matrix or data.frame.

Value

A numeric matrix with two columns.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

Examples

```
x = c(2.3, NA, -1.4, 0, 3.45)

find.na(x)
```

gapminder *Gapminder Data on Global Health, Income, and Population*

Description

The *gapminder* dataset provides global health, income, and population indicators for 195 countries over the period 1950–2019.

Usage

```
data(gapminder)
```

Format

The gapminder dataset, provided as a data frame, contains 13,650 rows and 8 columns (features) as follows:

- country: Country name.
- year: Calendar year of observation (1950–2019).
- gdp: Gross domestic product (GDP) in USD, based on World Bank data.
- life_expectancy: Average life expectancy at birth (in years).
- population: National population size.
- continent: Continent to which the country belongs.
- iso_alpha: ISO 3166-1 alpha-3 country code.
- world_group: A five-category geopolitical grouping of countries used for visualization, with levels *The West*, *Asia*, *Latin America*, *Africa*, and *Others*.

Details

For more information related to the dataset see:
https://www.gapminder.org/data/documentation/

Source

This dataset is originally from https://www.gapminder.org/resources/

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn, churn_mlc, churn_tel, adult, risk, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(gapminder)

str(gapminder)
```

---

house *house data set*

Description

the house dataset contains 6 features and 414 records. the target feature is *unit.price* and the remaining 5 variables are predictors.

Usage

```
data(house)
```

Format

the house dataset, as a data frame, contains 414 rows and 6 columns (variables/features). the 6 variables are:

- house.age: house age (numeric, in year).
- distance.to.MRT: distance to the nearest MRT station (numeric).
- stores.number: number of convenience stores (numeric).
- latitude: latitude (numeric).
- longitude: longitude (numeric).
- unit.price: house price of unit area (numeric).

Details

For more information related to the dataset see:
https://archive.ics.uci.edu/dataset/477/real+estate+valuation+data+set
https://www.kaggle.com/quantbruce/real-estate-price-prediction

Source

This dataset originally comes from the UCI repository of machine learning databases:
https://archive.ics.uci.edu/dataset/477/real+estate+valuation+data+set

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, risk, cereal, advertising, marketing, drug, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(house)
str(house)
```

---

house_price *house_price dataset*

Description

This data set contains 1460 rows and 81 columns (features). the "SalePrice" column is the target.

Usage

```
data(house_price)
```

Format

the house_price dataset, as a data frame, contains 1460 rows and 81 columns (variables/features).

Details

For more information related to the dataset see:
https://www.kaggle.com/datasets/lespin/house-prices-dataset

Source

This dataset comes from:
https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, risk, cereal, advertising, marketing, drug, house, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(house_price)
str(house_price)
```

`insurance` *insurance data set*

Description

the insurance dataset contains 7 features and 1338 records. the target feature is *charge* and the remaining 6 variables are predictors. This dataset is simulated on the basis of demographic statistics from the US Census Bureau.

Usage

```
data(insurance)
```

Format

the `insurance` dataset, as a data frame, contains 1338 rows (customers) and 7 columns (variables/features). the 7 variables are:

- `age`: age of primary beneficiary.
- `bmi`: body mass index, providing an understanding of body, weights that are relatively high or low relative to height, objective index of body weight (kg / m ^ 2) using the ratio of height to weight, ideally 18.5 to 24.9.
- `children`: Number of children covered by health insurance / Number of dependents.
- `smoker`: Smoking as a factor with 2 levels, yes, no.
- `gender`: insurance contractor gender, female, male.
- `region`: the beneficiary's residential area in the US, northeast, southeast, southwest, northwest.
- `charge`: individual medical costs billed by health insurance.

Details

For more information related to the dataset see:
https://www.kaggle.com/mirichoi0218/insurance

Source

This dataset comes from:
https://github.com/stedy/Machine-Learning-with-R-datasets

References

Brett Lantz (2019). Machine Learning with R: Expert techniques for predictive modeling. *Packt Publishing Ltd*.

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, risk, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, caravan, loan

Examples

```
data(insurance)
str(insurance)
```

---

kNN *k-Nearest Neighbour Classification*

Description

kNN is used to perform k-nearest neighbour classification for test set using training set. For each row of the test set, the k nearest (based on Euclidean distance) training set vectors are found. then, the classification is done by majority vote (ties broken at random). This function provides a formula interface to the class::knn() function of R package class. In addition, it allows normalization of the given data using the scaler function.

Usage

```
kNN(formula, train, test, k = 1, scaler = FALSE, type = "class", l = 0,
    use.all = TRUE, na.rm = FALSE)
```

Arguments

formula	a formula, with a response but no interaction terms. For the case of data frame, it is taken as the model frame (see model.frame).
train	data frame or matrix of train set cases.
test	data frame or matrix of test set cases.
k	number of neighbours considered.
scaler	a character with options FALSE (default), "minmax", and "zscore". Option "minmax" means no transformation. This option allows the users to use normalized version of the train and test sets for the kNN aglorithm.
type	either "class" (default) for the predicted class or "prob" for model confidence values.
l	minimum vote for definite decision, otherwise doubt. (More precisely, less than k-l dissenting votes are allowed, even if k is increased by ties.)
use.all	controls handling of ties. If true, all distances equal to the kth largest are included. If false, a random selection of distances equal to the kth is chosen to use exactly k neighbours.
na.rm	a logical value indicating whether NA values in x should be stripped before the computation proceeds.

Value

When type = "class" (default), a factor vector is returned, in which the doubt will be returned as NA. When type = "prob", a matrix of confidence values is returned (one column per class).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

References

Ripley, B. D. (1996) *Pattern Recognition and Neural Networks.* Cambridge.
Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S.* Fourth edition. Springer.

See Also

kNN, scaler

Examples

```
data(risk)

train = risk[1:100, ]
test  = risk[  101, ]

kNN(risk ~ income + age, train = train, test = test)
```

---

kNN.plot *Visualizing the Optimal Number of k*

Description

Visualizing the Optimal Number of k for k-Nearest Neighbour (kNN) algorithm based on accuracy or Mean Square Error (MSE).

Usage

```
kNN.plot(formula, train, test = NULL, ratio = c(0.7, 0.3), k.max = 10,
         scaler = FALSE, base = "accuracy", reference = NULL, cutoff = NULL,
         type = "class", report = FALSE, set.seed = NULL, ...)
```

Arguments

formula	a formula, with a response but no interaction terms. For the case of data frame, it is taken as the model frame (see model.frame).
train	data frame or matrix of train set cases.
test	Data frame or matrix containing the test set observations. If NULL, the train data are partitioned according to ratio.

ratio	Numeric vector of length 1 or 2 specifying the proportions used by partition() to split the train data into training and validation sets.
k.max	the maximum number of neighbors to consider can either be a single value, with a minimum of 2, or a vector representing a range of values k.
scaler	a character with options FALSE (default), "minmax", and "zscore". Option "minmax" means no transformation. This option allows the users to use normalized version of the train and test sets for the kNN aglorithm.
base	base measurement: accuracy (default), error, or MSE for Mean Square Error.
reference	a factor of classes to be used as the true results.
cutoff	cutoff value for the case that the output of knn algorithm is vector of probabilites.
type	either "class" (default) for the predicted class or "prob" for model confidence values.
report	a character with options FALSE (default) and TRUE. Option TRUE reports the values of the base measurement.
set.seed	a single value, interpreted as an integer, or NULL.
...	options to be passed to kNN().

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

References

Ripley, B. D. (1996) *Pattern Recognition and Neural Networks.* Cambridge.
Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S.* Fourth edition. Springer.

See Also

kNN, scaler

Examples

```
data(risk)

partition_risk <- partition(data = risk, ratio = c(0.6, 0.4))

train <- partition_risk$part1
test  <- partition_risk$part1

kNN.plot(risk ~ income + age, train = train, test = test)
kNN.plot(risk ~ income + age, train = train, test = test, base = "error")
```

`loan` *Loan Application and Approval Data*

Description

A dataset containing information on loan applicants and their financial profiles, including demographic characteristics, employment status, income, loan details, credit score, asset values, and loan approval outcome.

Usage

```
data(loan)
```

Format

A data frame with 4269 observations and 13 variables:

loan_id Unique identifier for each loan application; not intended as a predictor in modeling.

no_of_dependents Number of dependents of the applicant.

education Education level of the applicant: `"graduate"` or `"not-graduate"`.

self_employed Whether the applicant is self-employed: `"yes"` or `"no"`.

income_annum Annual income of the applicant.

loan_amount Requested loan amount.

loan_term Loan term.

cibil_score Applicant's CIBIL credit score.

residential_assets_value Value of the applicant's residential assets.

commercial_assets_value Value of the applicant's commercial assets.

luxury_assets_value Value of the applicant's luxury assets.

bank_asset_value Value of the applicant's bank assets.

loan_status Loan application outcome: `"approved"` or `"rejected"`.

Details

This dataset was obtained from Kaggle and renamed `loan` for inclusion in the **liver** package. It can be used to illustrate methods for classification, exploratory data analysis, and predictive modeling in R.

Source

https://www.kaggle.com/datasets/architsharma01/loan-approval-prediction-dataset

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

mortgage, bank, churn_mlc, churn, churn_tel, adult, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(loan)
str(loan)
summary(loan)

table(loan$loan_status)
```

---

mae *Mean Absolute Error (MAE)*

Description

Computes mean absolute error.

Usage

```
mae(pred, actual, weight = 1, na.rm = FALSE)
```

Arguments

pred	a numerical vector of estimated values.
actual	a numerical vector of actual values.
weight	a numerical vector of weights the same length as pred.
na.rm	a logical value indicating whether NA values in pred should be stripped before the computation proceeds.

Value

the computed mean squared error (numeric value).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

See Also

mse

Examples

```
pred   = c(2.3, -1.4, 0, 3.45)
actual = c(2.1, -0.9, 0, 2.99)

mae(pred, actual)
```

`marketing` *marketing data set*

Description

the marketing dataset contains 8 features and 40 records as 40 days that report how much we spent, how many clicks, impressions and transactions we got, whether or not a display campaign was running, as well as our revenue, click-through-rate and conversion rate. the target feature is *revenue* and the remaining 7 variables are predictors.

Usage

```
data(marketing)
```

Format

the `marketing` dataset, as a data frame, contains 40 rows and 8 columns (variables/features). the 8 variables are:

- `spend`: daily send of money on PPC (apy-per-click).
- `clicks`: number of clicks on for that ad.
- `impressions`: amount of impressions per day.
- `display`: whether or not a display campaign was running.
- `transactions`: number of transactions per day.
- `click.rate`: click-through-rate.
- `conversion.rate`: conversion rate.
- `revenue`: daily revenue.

Details

For more information related to the dataset see:
https://github.com/chrisBow/marketing-regression-part-one

Source

This dataset comes from:
https://github.com/chrisBow/marketing-regression-part-one

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, risk, cereal, advertising, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(marketing)
str(marketing)
```

`minmax` *Min-Max scaling of numerical variables*

Description

Performs Min-Max tranformation for numerical variables.

Usage

```
minmax(x, col = "auto", min = NULL, max = NULL, na.rm = FALSE)
```

Arguments

`x`	a numerical `vector`, `matrix` or `data.frame`.
`col`	a character vector of column names or indices. If `"auto"`, all numeric columns will be transformed. If `"all"`, all columns will be transformed.
`min`	a numerical value or vector indicating the minimum value(s) to use for Min-Max tranformation; if NULL, the default is based on `x`.
`max`	a numerical value or vector indicating the maximum value(s) to use for Min-Max tranformation; if NULL, the default is based on `x`.
`na.rm`	a logical value indicating whether NA values in `x` should be stripped before the computation proceeds.

Value

transformed version of `x`.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

See Also

`scaler`, `zscore`

Examples

```
x = c(2.3, -1.4, 0, 3.45)

minmax(x)

minmax(x, min = 0, max = 1)
```

`mortgage` *Mortgage data set*

Description

The *mortgage* dataset contains 850 records and 8 variables. The target variable is *risk*, a factor with two levels, `"low"` and `"high"`. The remaining seven variables serve as predictors. The dataset was simulated to represent a realistic mortgage application setting.

Usage

```
data(mortgage)
```

Format

A data frame with 850 rows (applicants) and 8 variables:

- `age`: Age in years.
- `income`: Annual income.
- `savings`: Total savings.
- `employment_status`: A factor with levels `"permanent"`, `"temporary"`, `"self_employed"`, and `"unemployed"`.
- `credit_history`: A factor with levels `"poor"`, `"average"`, and `"good"`.
- `debt_level`: A factor with levels `"low"`, `"medium"`, and `"high"`.
- `loan_amount`: Requested loan amount.
- `risk`: A factor with levels `"low"` and `"high"`.

Details

The dataset was generated using a hybrid latent simulation approach. Continuous variables were simulated with dependence, and categorical variables were derived from latent scores to create realistic relationships among applicant characteristics, financial indicators, and mortgage risk.

Source

Simulated data generated for illustration and teaching purposes.

References

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(mortgage)
str(mortgage)
```

`mse` *Mean Squared Error (MSE)*

Description

Computes mean squared error.

Usage

```
mse(pred, actual, weight = 1, na.rm = FALSE)
```

Arguments

`pred`	a numerical vector of estimated values.
`actual`	a numerical vector of actual values.
`weight`	a numerical vector of weights the same length as `pred`.
`na.rm`	a logical value indicating whether NA values in `pred` should be stripped before the computation proceeds.

Value

the computed mean squared error (numeric value).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

See Also

`mae`

Examples

```
pred   = c(2.3, -1.4, 0, 3.45)
actual = c(2.1, -0.9, 0, 2.99)

mse(pred, actual)
```

one.hot *One Hot Encoder*

Description

One-Hot-Encode unordered factor columns of a data.frame, matrix, or data.table, using the mltools::one_hot() mltools::one_hot function.

Usage

```
one.hot(data, cols = "auto", sparsifyNAs = FALSE, naCols = FALSE,
                  dropCols = FALSE, dropUnusedLevels = FALSE)
```

Arguments

data	a numerical vector, matrix, data.frame, or data.table.
cols	a character vector of column names or indices to one-hot-encode. If "auto", all unordered factor columns will be one-hot-encoded.
sparsifyNAs	a logical value indicating whether to converte NAs to 0s.
naCols	a logical value indicating whether to create a separate column for NAs.
dropCols	a logical value indicating whether to drop the original columns which are one-hot-encoded.
dropUnusedLevels	a logical value indicating whether to drop unused factor levels.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

See Also

scaler

Examples

```
data(risk)
str(risk)

risk_one_hot <- one.hot(risk, cols = "auto")
str(risk_one_hot)
```

partition *Partition the data*

Description

Randomly partitions the data (primarly intended to split into "training" and "test" sets) according to the supplied probabilities.

Usage

```
partition(data, ratio = c(0.7, 0.3), set.seed = NULL)
```

Arguments

data	an $(n \times p)$ matrix or a data.frame.
ratio	a numerical vector in range of [0, 1].
set.seed	a single value, interpreted as an integer, or NULL.

Value

a list which includes the data partitions.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

Examples

```
data(iris)

partition(data = iris, ratio = c(0.7, 0.3))
```

prop.conf *Confdidence interval for proportion*

Description

Compute a confidence interval for the proportion of a response variable using the normal distribution.

Usage

```
prop.conf(x, n, conf = 0.95, ...)
```

Arguments

x	a vector of counts of successes, a one-dimensional table with two entries, or a two-dimensional table (or matrix) with 2 columns, giving the counts of successes and failures, respectively.
n	a vector of counts of trials; ignored if x is a matrix or a table.
conf	confidence level of the interval.
...	further arguments to be passed to prop.test.

Value

A vector with two values: lower and upper confidence limits for the proportion of the response variable.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

Examples

```
data(churn_mlc)

prop.conf(table(churn_mlc$churn), conf = 0.9)
```

---

purchase_intention *Online Shopper Purchase Intention Data*

Description

A dataset containing session-level information from an e-commerce website, including page visit counts, time spent in different page categories, Google Analytics metrics, visitor characteristics, and a binary outcome indicating whether the session ended in a purchase. The dataset can be used to illustrate binary classification, exploratory data analysis, model comparison, and supervised learning methods in R.

Usage

```
data(purchase_intention)
```

Format

A data frame with 12330 observations and 18 variables:

administrative Number of administrative pages visited during the session.

administrative_duration Total time spent on administrative pages during the session.

informational Number of informational pages visited during the session.

informational_duration Total time spent on informational pages during the session.

product_related Number of product-related pages visited during the session.

product_related_duration Total time spent on product-related pages during the session.

bounce_rates Average bounce rate associated with the visited pages.

exit_rates Average exit rate associated with the visited pages.

page_values Average page value for pages visited before a completed transaction.

special_day Closeness of the session date to a special shopping day, scaled between 0 and 1.

month Month of the session.

operating_systems Visitor operating system, recorded as a categorical factor.

browser Visitor browser, recorded as a categorical factor.

region Visitor region, recorded as a categorical factor.

traffic_type Traffic source type, recorded as a categorical factor.

visitor_type Visitor type: `"New_Visitor"`, `"Returning_Visitor"`, or `"Other"`.

weekend Whether the session occurred on a weekend: `"no"` or `"yes"`.

revenue Whether the session ended in a purchase: `"no"` or `"yes"`.

Details

This dataset was obtained from the UCI Machine Learning Repository and renamed `purchase_intention` for inclusion in the **liver** package. It contains session-level records from an online shopping website and is well suited for illustrating modern binary classification problems in which the goal is to predict whether a browsing session will end in a purchase.

The predictors combine behavioral measures such as page visit counts and time spent on different types of pages with summary metrics such as `bounce_rates`, `exit_rates`, and `page_values`, as well as visitor and session characteristics including `month`, `visitor_type`, `traffic_type`, and `weekend`. The outcome variable `revenue` indicates whether the session resulted in a completed transaction.

The dataset is particularly useful for demonstrating classification workflows such as partitioning data into training and test sets, fitting logistic regression, Naive Bayes, k-nearest neighbors, and tree-based models, and evaluating predictive performance using confusion matrices, ROC curves, and AUC.

Source

UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset

Sakar, C. and Kastro, Y. (2018). *Online Shoppers Purchasing Intention Dataset*. doi:10.24432/C5F88Q

References

Sakar, C. O., Polat, S. O., Katircioglu, M., and Kastro, Y. (2019). Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks. *Neural Computing and Applications*, 31, 6893–6908. doi:10.1007/s0052101835230

Reza Mohammadi (2025). *Data Science Foundations and Machine Learning with R: From Data to Decisions*. https://book-data-science-r.netlify.app

See Also

mortgage, bank, churn_mlc, churn, churn_tel, adult, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(purchase_intention)

str(purchase_intention)

summary(purchase_intention)
```

---

red_wines *Red wines data set*

Description

the red_wines datasets are related to red variants of the Portuguese "Vinho Verde" wine. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

the dataset can be viewed as classification or regression tasks. the classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant. So it could be interesting to test feature selection methods.

Usage

```
data(red_wines)
```

Format

the red_wines dataset, as a data frame, contains 1599 rows and 12 columns (variables/features). the 12 variables are:

Input variables (based on physicochemical tests):

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates

- `alcohol`
  Output variable (based on sensory data)
- `quality`: score between 0 and 10.

Details

For more information related to the dataset see the UCI Machine Learning Repository:
https://archive.ics.uci.edu/dataset/186/wine+quality

Source

This dataset comes from the UCI repository of machine learning databases:
https://archive.ics.uci.edu/dataset/186/wine+quality

References

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4), 547-553.

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, risk, cereal, advertising, marketing, drug, house, house_price, white_wines, insurance, caravan, loan

Examples

```
data(red_wines)
str(red_wines)
```

`risk` *Risk data set*

Description

The *risk* dataset contains 246 records and 6 variables. The target variable is *risk*, a factor with two levels ("good risk" and "bad risk"). The remaining five variables serve as predictors. The dataset was simulated to reflect a realistic real-world scenario.

Usage

```
data(risk)
```

Format

the risk dataset, as a data frame, contains 246 rows (customers) and 6 columns (variables/features). the 6 variables are:

- age: age in years.
- marital: A factor with levels "single", "married", and "other".
- income: yearly income.
- mortgage: A factor with levels "yes" and "no".
- nr_loans: Number of loans that constomers have.
- risk: A factor with levels "good risk" and "bad risk".

References

Larose, D. T. and Larose, C. D. (2014). Discovering knowledge in data: an introduction to data mining. *John Wiley & Sons*.

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(risk)
str(risk)
```

---

scaler *Feature scaling*

Description

Performs feature scaling such as Z-score and min-max scaling.

Usage

```
scaler(x, scale = c("minmax", "zscore"), col = "auto",
       par1 = NULL, par2 = NULL, na.rm = FALSE)
```

Arguments

x	a numerical vector, a matrix or a data.frame.
scale	a transfer for x.
col	a character vector of column names or indices. If "auto", all numeric columns will be transformed. If "all", all columns will be transformed.
par1	a numerical value or vector that for the case scale = "minmax" indicating the maximum value(s) and for the case scale = "zscore" indicating the mean value(s).
par2	a numerical value or vector that for the case scale = "minmax" indicating the maximum value(s) and for the case scale = "zscore" indicating the sd value(s).
na.rm	a logical value indicating whether NA values in x should be stripped before the computation proceeds.

Value

transformed version of x.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

See Also

zscore, minmax

Examples

```
x = c(2.3, -1.4, 0, 3.45)

scaler(x, scale = "minmax")

scaler(x, scale = "zscore")
```

---

skewness *Skewness*

Description

Computes the skewness for each field.

Usage

```
skewness(x, na.rm = FALSE)
```

Arguments

x	a numerical vector, matrix or data.frame.
na.rm	a logical value indicating whether NA values in x should be stripped before the computation proceeds.

Value

A numeric vector of skewness values.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

Examples

```
x = c(2.3, -1.4, 0, 3.45)

skewness(x)
```

`skim` *Skim a data frame to get useful summary statistics*

Description

`skim()` provides an overview of a data frame asan alternative to `summary()`. This function is a wrapper for the `skimr::skim()` function of R package `skimr`.

Usage

```
skim(data, hist = TRUE, ...)
```

Arguments

`data`	a data frame or matrix.
`hist`	Logical: TRUE (default) to report the histogram of each variable.
`...`	columns to select for skimming. the default is to skim all columns.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

See Also

`summary()`

Examples

```
data(risk)

skim(risk)
```

t_conf *Confdidence interval for mean*

Description

Compute a confidence interval for the mean of a response variable using the t-distribution.

Usage

```
t_conf(x, conf = 0.95, ...)
```

Arguments

x	a (non-empty) numeric vector of data values.
conf	confidence level of the interval.
...	further arguments to be passed to `t.test`.

Value

A vector with two values: lower and upper confidence limits for the mean of the response variable.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

Examples

```
data(churn_mlc)

t_conf(churn_mlc$customer_calls, conf = 0.9)
```

white_wines *White wines data set*

Description

the white_wines datasets are related to white variants of the Portuguese "Vinho Verde" wine. Due to privacy and logistic issues, only physicochemical (inputs) and sensory (the output) variables are available (e.g. there is no data about grape types, wine brand, wine selling price, etc.).

the dataset can be viewed as classification or regression tasks. the classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones). Outlier detection algorithms could be used to detect the few excellent or poor wines. Also, we are not sure if all input variables are relevant. So it could be interesting to test feature selection methods.

Usage

```
data(white_wines)
```

Format

the white_wines dataset, as a data frame, contains 4898 rows and 12 columns (variables/features). the 12 variables are:

Input variables (based on physicochemical tests):

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates
- alcohol

Output variable (based on sensory data)

- quality: score between 0 and 10.

Details

For more information related to the dataset see the UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/186/wine+quality

Source

This dataset comes from the UCI repository of machine learning databases: https://archive.ics.uci.edu/dataset/186/wine+quality

References

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4), 547-553.

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

bank, churn_mlc, churn, churn_tel, adult, risk, cereal, advertising, marketing, drug, house, house_price, red_wines, insurance, caravan, loan

Examples

```
data(white_wines)
str(white_wines)
```

`wholesale_customers` *Wholesale Customer Spending Data*

Description

A dataset containing annual spending information for clients of a wholesale distributor, along with the customer's sales channel and geographic region. The dataset can be used to illustrate customer segmentation, clustering, exploratory data analysis, and unsupervised learning methods in R.

Usage

```
data(wholesale_customers)
```

Format

A data frame with 440 observations and 8 variables:

fresh Annual spending on fresh products (in monetary units).

milk Annual spending on milk products (in monetary units).

grocery Annual spending on grocery products (in monetary units).

frozen Annual spending on frozen products (in monetary units).

detergents_paper Annual spending on detergents and paper products (in monetary units).

delicassen Annual spending on delicatessen products (in monetary units).

channel Customer sales channel: `"Horeca"` or `"Retail"`.

region Customer region: `"Lisbon"`, `"Oporto"`, or `"Other"`.

Details

This dataset was obtained from the UCI Machine Learning Repository and renamed `wholesale_customers` for inclusion in the **liver** package. It refers to clients of a wholesale distributor and records their annual spending in several product categories. The dataset is well suited for illustrating methods for clustering, customer profiling, and multivariate data exploration.

In clustering applications, the numerical spending variables are typically used to define the clusters, while `channel` and `region` can be used afterward to help interpret the resulting customer groups.

Source

UCI Machine Learning Repository: https://archive.ics.uci.edu/dataset/292/wholesale+customers

References

B. Jaya Lakshmi, K. B. Madhuri, and M. Shashi (2017). An Efficient Algorithm for Density Based Subspace Clustering with Dynamic Parameter Setting. *International Journal of Information Technology and Computer Science*, 9(6), 27–33. doi:10.5815/ijitcs.2017.06.04

Reza Mohammadi (2025). Data Science Foundations and Machine Learning with R: From Data to Decisions. https://book-data-science-r.netlify.app.

See Also

mortgage, bank, churn_mlc, churn, churn_tel, adult, cereal, advertising, marketing, drug, house, house_price, red_wines, white_wines, insurance, caravan, loan

Examples

```
data(wholesale_customers)

str(wholesale_customers)

summary(wholesale_customers)
```

---

z.conf *Confdidence interval for mean using z-distribution*

Description

Compute a confidence interval for the mean of a response variable using the z-distribution.

Usage

```
z.conf(x, sigma = NULL, conf = 0.95)
```

Arguments

x	a (non-empty) numeric vector of data values.
sigma	the population standard deviation. If NULL, the sample standard deviation is used. This is useful when the population standard deviation is known, otherwise it should be left as NULL.
conf	confidence level of the interval.

Value

A vector with two values: lower and upper confidence limits for the mean of the response variable.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

Examples

```
data(churn_mlc)

z.conf(x = churn_mlc$customer_calls, conf = 0.9)
```

`zscore` *Z-score scaling of numerical variables*

Description

Performs Z-score tranformation for numerical variables.

Usage

```
zscore(x, col = "auto", mean = NULL, sd = NULL, na.rm = FALSE)
```

Arguments

`x`	a numerical `vector`, `matrix` or `data.frame`.
`col`	a character vector of column names or indices. If `"auto"`, all numeric columns will be transformed. If `"all"`, all columns will be transformed.
`mean`	a numerical value or vector indicating the `mean` to use for Z-score calculation; if NULL, the default is the mean of `x`.
`sd`	a numerical value or vector indicating the standard deviation(s) to use for Z-score calculation; if NULL, the default is the standard deviation of `x`.
`na.rm`	a logical value indicating whether NA values in `x` should be stripped before the computation proceeds.

Value

transformed version of `x`.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Kevin Burke <kevin.burke@ul.ie>

See Also

`scaler`, `minmax`

Examples

```
x = c(2.3, -1.4, 0, 3.45)

zscore(x)
zscore(x, mean = 1, sd = 2)
```

Index